



**Facultad
de
Ciencias**

**Reconstrucción del perfil de masa de
cúmulos de galaxias a través del efecto de
lente gravitatoria y un método de
maximización de la aleatoriedad de
imágenes**

**(Galaxy cluster's mass reconstruction using
gravitational lensing and a method based on
the maximization of image randomness)**

**Trabajo de Fin de Grado
para acceder al**

GRADO EN FÍSICA

Autor: Irati Lizaso Berrueta

Director: Diego Herranz Muñoz

Junio - 2021

Agradecimientos

Me gustaría reconocer el gran trabajo de Andrea Petri en el desarrollo de su *software* LensTools, y a él personalmente por haber atendido mis preguntas sobre los detalles del programa.

También quiero agradecer al director de este trabajo, por servir de ayuda y guía en mi aprendizaje y por haberme proporcionado la motivación e impulso necesarios para querer adentrarme en un proyecto tan interesante.

Y en particular quiero dedicar este pequeño espacio a mi familia y amigos, por haber escuchado durante horas y horas mis reflexiones sobre elipses, lensing y entropía.

Resumen

En el presente trabajo se busca estudiar el efecto de lente gravitatoria y se explora una nueva técnica de reconstrucción de masas basada en este efecto. Partiendo de un modelo paramétrico elíptico de la distribución de masa de un cúmulo de galaxias actuando como lente, y debido a que el alineamiento que produce el *lensing* gravitatorio sobre galaxias de fondo disminuye su aleatoriedad, el método propuesto consiste en invertir la ecuación de lente y medir la aleatoriedad de las imágenes de-lensadas mediante el cálculo de la entropía de Shannon, después de lo cual se obtendrán los parámetros del cúmulo que maximicen dicha aleatoriedad y se estimará la masa total del cúmulo.

En concreto, tras un repaso del marco teórico del efecto de lente gravitatoria, se implementará un código que utilice el método descrito para la reconstrucción del perfil de masa de un cúmulo de galaxias, pudiendo finalmente comparar los valores de los parámetros estimados con los valores reales y estudiar la viabilidad de esta técnica de reconstrucción de masas.

Índice general

1. Introducción	2
1.1. Introducción a las lentes gravitatorias	4
1.1.1. Aplicaciones	7
1.2. El formalismo de lente	9
1.2.1. Ecuación de lente	9
1.2.2. Ángulo de deflexión y potencial de <i>lensing</i>	10
2. Weak lensing de cúmulos de galaxias sobre galaxias de fondo	13
2.1. Simulación de la lente gravitatoria	13
2.1.1. Perfil de masa de los cúmulos de galaxias: distribución de Navarro, Frenk y White	14
2.1.2. Modelo de potencial elíptico NFW	14
2.1.3. Simulaciones	15
2.2. Simulación de galaxias	18
2.3. Lensado y de-lensado de galaxias	18
2.3.1. Resultados	19
3. Estimación de la lente a través de la maximización de la aleatoriedad	22
3.1. Estudio de la aleatoriedad	22
3.1.1. Entropía global de Shannon	22
3.2. Maximización de la aleatoriedad: determinación de la distribución de masa	24
3.3. Estimación de la masa de un cúmulo	27
3.4. Discusión del método y futuras investigaciones	28
4. Conclusiones	31
A. Código	34
Bibliografía	48

Capítulo 1

Introducción

Desde las primeras ideas sobre la deflexión gravitatoria de la luz, ya desde el siglo XVIII, y sobre todo a partir de principios del siglo XX, cuando varias investigaciones como las de Albert Einstein y Fritz Zwicky llevaron al desarrollo de la teoría de las lentes gravitatorias, este fenómeno se ha convertido en uno de los principales objetos de investigación de la astrofísica moderna. A día de hoy existe una considerable cantidad de literatura publicada sobre *lensing*¹ gravitatorio, lo que revela el impacto que ha tenido en la comunidad científica y su potencial como herramienta para estudiar un gran número de cuestiones fundamentales en la astrofísica.

Una de sus diversas aplicaciones es la estimación de masas, en particular la reconstrucción de distribuciones de masa de cúmulos de galaxias. Es el hecho de que el efecto de lente gravitacional dependa únicamente de la distribución total de masa, y no de otros factores como la luminosidad, la composición o el estado dinámico en el que se encuentra el cúmulo (factores de los cuales sí dependen otros métodos de estimación de masas), lo que hace que el *lensing* resulte una técnica excelente para el estudio de cúmulos de galaxias.

Estudiar los cúmulos de galaxias es muy importante en la astrofísica. Los cúmulos de galaxias son las estructuras ligadas más masivas del Universo, y están compuestos de un 10 % de materia luminosa en contraste con un 90 % de materia oscura [1]. Desde el postulado de su existencia en 1930 a causa de varias observaciones indirectas [2], la materia oscura ha supuesto uno de los enigmas más fascinantes de la cosmología moderna. Debido a que este tipo de materia no bariónica no interacciona con la radiación electromagnética solo puede detectarse por sus efectos gravitatorios, y esto hace que su detección suponga aún un reto para la comunidad científica. Lo que sí puede afirmarse a día de hoy es que constituye un 27 % del Universo (frente a la materia bariónica que supone tan solo un 5 %) [3], por lo que este tipo de materia resulta especialmente relevante por el papel que desempeña sobre la estructura a gran escala del Universo.

¹El término *lensing* se refiere al efecto de lente gravitatoria. En adelante se usará este anglicismo debido a que no existe una traducción compacta análoga en castellano.

De esta manera, los cúmulos de galaxias se convierten en los mejores laboratorios naturales para estudiar la materia oscura.

El principal objetivo del presente trabajo será por tanto estimar la masa de un cúmulo de galaxias, lo que puede permitir aprender más acerca de la materia oscura y por ende sobre la formación y evolución de la estructura a gran escala del Universo.

Dentro del campo de las lentes gravitatorias existen varios métodos para reconstruir la masa de un cúmulo. Uno de los métodos más utilizados es el de Kaiser-Squires [4, 5], basado en medir las distorsiones y las magnificaciones causadas por el efecto *lensing* y calcular con ello la densidad superficial de masa de la lente. En contraste con este método “directo”, existen también los llamados métodos inversos [6, 7], que consisten en calcular las distorsiones y magnificaciones que generaría una lente hasta encontrar una que se ajuste bien con los datos observados. Estos últimos suelen trabajar con ajustes de máxima probabilidad o de máxima entropía [8, 9], donde teniendo unos datos con cierto ruido, se opta por darle a estos datos aquella distribución de probabilidad con mayor entropía. Para aplicar esta técnica es necesario tener de antemano cierta información sobre el cúmulo. Como alternativa a los métodos no paramétricos (o de *free-form*), con los que se puede llegar a obtener picos de densidad cuya interpretación no siempre es segura (pueden deberse a una concentración de masa o a un error debido a la elipticidad de las galaxias), a menudo conviene utilizar modelos paramétricos para ajustar los datos. Los modelos de lente suelen ser esféricos o elípticos, siendo éste último el caso que nos ocupa.

En este proyecto se propone un nuevo método para reconstruir distribuciones de masa de cúmulos de galaxias. Esta técnica se basa en la idea de que *lensar*² una distribución de galaxias disminuye su aleatoriedad, las ordena, por lo que se puede estimar la distribución de masa de la lente maximizando la aleatoriedad de la imagen de las galaxias al invertir las ecuaciones de lente, es decir, al *de-lensar* las galaxias. Esta aleatoriedad se cuantificará utilizando la medida de entropía de Shannon. He de señalar que este método no guarda similitud con el problema inverso de maximización de entropía mencionado, a pesar de haberlos referido de forma parecida. A partir de este punto, al hablar de maximización de entropía se estará haciendo referencia a la aleatoriedad del método implementado.

Por lo tanto, en este trabajo se buscará implementar un código que tras generar galaxias de fondo aleatorias y cúmulos de galaxias con un modelo paramétrico siguiendo un perfil de densidad NFW elíptico, maximice la aleatoriedad de las galaxias *de-lensadas* y determine los parámetros de la lente que la maximiza, pudiendo así obtener la densidad superficial de masa y una estimación de la masa del cúmulo.

²Con *lensar* un fondo nos referimos a obtener el efecto de lente del fondo, y con *de-lensar* a deshacer o invertir el efecto lente. De ahora en adelante se utilizarán estos términos por comodidad.

Este trabajo viene estructurado como sigue: en el Capítulo 1 se hará una breve introducción tanto histórica como fenomenológica y se deducirán las principales ecuaciones del *lensing*. Tras esto el trabajo se dividirá en dos partes principales, el Capítulo 2, donde se realizará la simulación del lensado y de-lensado de galaxias por cúmulos previamente generados, y el Capítulo 3, donde se implementará la maximización de la aleatoriedad y se estimará la masa del cúmulo de galaxias, comentando los resultados de aplicar este método y proponiendo posibles desarrollos para futuras investigaciones. Finalmente, en el Capítulo 4 se proporciona un breve resumen de las conclusiones obtenidas.

1.1. Introducción a las lentes gravitatorias

El *lensing* gravitatorio es un área de gran interés en la astrofísica. Las primeras ideas comenzaron a surgir con la teoría gravitatoria de Newton; siguiendo el modelo corpuscular de la luz, el campo gravitatorio podría influir sobre los rayos de luz, desviándolos así de su trayectoria rectilínea. Astrónomos como Soldner (1804) o Laplace (1795) hicieron diversos cálculos utilizando el Sol como lente deflectora [10], concluyendo que un rayo de luz propagándose en el campo gravitatorio de una masa esférica M sufriría una desviación con un ángulo $\hat{\alpha}_N = 2GM/(c^2\xi)$ [3], donde G es la constante gravitatoria universal, c es la velocidad de la luz y ξ es el parámetro de impacto del rayo de luz incidente.

No es hasta un siglo después cuando empieza a darse importancia al efecto *lensing*, cuando Einstein publica una detallada derivación de las ecuaciones de lente [11]. Esta vez desde el punto de vista de la Relatividad General y partiendo del principio de equivalencia, Einstein llega a la expresión para el ángulo de deflexión $\hat{\alpha} = 4GM/(c^2\xi)$, que resulta ser el doble que el valor newtoniano $\hat{\alpha}_N$. Los resultados de Einstein fueron confirmados en 1919 en un eclipse solar total, observando las estrellas proyectadas cerca de la superficie solar y midiendo la desviación de sus posiciones reales [12], lo que supuso un gran éxito para la Relatividad General.

A partir de este punto el efecto *lensing* tomó una relevancia considerable, y la comunidad científica comenzó a utilizar el concepto de “lente gravitacional” para referirse a la masa deflectora. De esta forma, un sistema de lente está formado por una fuente de luz (*source*), la lente gravitatoria y el observador.

La Fig. 1.1 ilustra el sistema de *lensing* gravitatorio mencionado. Puede verse que los rayos de luz procedentes de la fuente se curvan en su trayectoria hasta el observador, debido a la lente gravitatoria. En este sistema hay un perfecto alineamiento entre la fuente, la lente y el observador, situación que en ciertas circunstancias (con los

³El subíndice N en la expresión del ángulo de deflexión denota que éste ha sido calculado usando la teoría newtoniana de la gravedad.

elementos a distancias concretas) podría producir el llamado anillo de Einstein. Cuando los tres elementos se desalinean un poco, el anillo se convierte en arcos (ver Fig. 1.2).

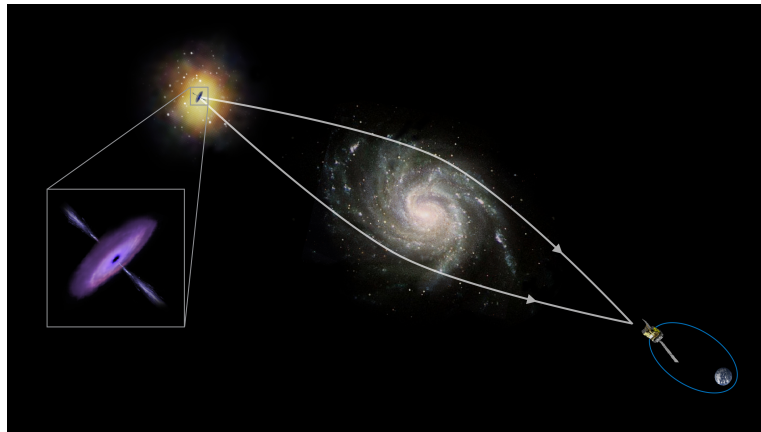


Figura 1.1. Un cuásar como fuente (arriba a la izquierda), cuyos rayos de luz se curvan por la galaxia a modo de lente gravitatoria, y llegan al observador terrestre. Fuente: ESA/ATG medialab (http://www.esa.int/ESA_Multimedia/Images/2015/07/Gravitational_lensing).

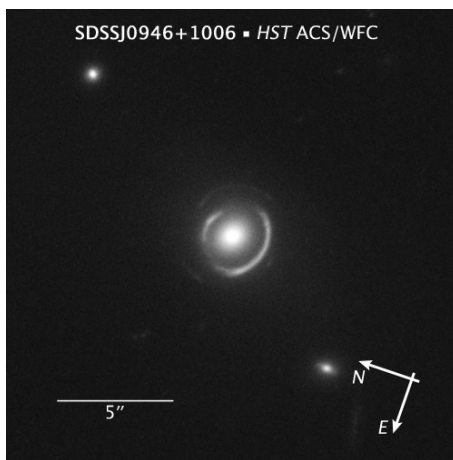


Figura 1.2. Imagen del sistema de lente gravitatoria SDSSJ0946+1006. El sistema consta de una galaxia elíptica *lensando* la luz de dos galaxias lejanas a distintas distancias y casi perfectamente alineadas, por lo que forman un anillo de Einstein casi completo. Fuente: NASA / ESA, Hubble Space Telescope's Advanced Camera for Surveys (<https://hubblesite.org/contents/media/images/2008/04/2245-Image.html>).

Pero hasta entonces solo se había contemplado el *lensing* estelar dentro de nuestra galaxia. Sin embargo, Einstein concluyó (1936) que las separaciones angulares de las imágenes del efecto *lensing* serían demasiado pequeñas como para poder observarlas, debido a que las masas estelares eran muy pequeñas. En 1937, Zwicky propuso utilizar galaxias como lentes, ya que las separaciones angulares, de esta manera, serían suficientemente grandes como para resolverlas con telescopios. Más adelante, con el descubrimiento de los cuásares, se abrió una nueva puerta para el área del *lensing* gravitacional y comenzaron a observarse un gran número de objetos lensados en el Universo. Se observó, entre otras cosas, que un objeto lensado puede llegar a tener

múltiples imágenes, como se aprecia en la Fig. 1.3, con los rayos de luz tomando distintos caminos hasta el observador (esto predice el carácter no lineal de la ecuación de lente, que se verá más adelante).

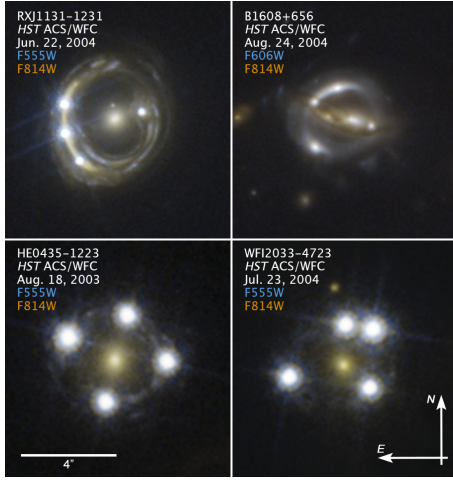


Figura 1.3. Cuatro sistemas de cuásares (RXJ1131-1231, B1608+656, HE0435-1223 y WFI2033-4723) lensados por galaxias. Cada cuásar tiene cuatro imágenes, formadas a causa del campo gravitatorio de la galaxia supermasiva. Fuente: NASA / ESA, Hubble Space Telescope (<https://hubblesite.org/contents/media/images/2020/04/4607-Image.html>).

Sin embargo, con el paso de los años se ha visto que no todos los efectos del *lensing* gravitatorio son tan fácilmente detectables. Dependiendo de ciertas características del sistema, clasificamos las lentes gravitacionales en tres grupos: *strong lensing* o régimen fuerte, *weak lensing* o régimen débil, y *microlensing*. Esta distinción está basada en el nivel de distorsión de las imágenes.

Por un lado, los regímenes fuerte y débil están separados por un valor crítico de la densidad de masa superficial de la lente, pero ambos se dan con galaxias o cúmulos de galaxias a modo de lente, debido a que son éstos los casos en los que se obtienen densidades de masa suficientemente altas [13]. Cuando un sistema está perfectamente alineado (o muy cerca de estarlo), su eje óptico pasa por el centro de la lente, que será donde mayor densidad de masa superficial haya, y esto lleva a un efecto de *strong lensing*. Este régimen genera imágenes notorias como son los mencionados anillos de Einstein o las imágenes múltiples. No obstante, no siempre encontramos sistemas alineados, lo que significa que el eje óptico no pasa exactamente por el centro de la lente (ya sea galaxia o cúmulo de galaxias) y tiene por tanto una densidad de masa superficial menor que el valor crítico necesario. En este caso nos encontramos en el régimen de *weak lensing*, donde el campo gravitatorio no es suficientemente intenso como para crear múltiples imágenes, y solo vemos una ligera distorsión de las imágenes, detectable únicamente mediante métodos estadísticos [14]. La diferencia entre regímenes fuerte y débil viene ilustrada en la Fig. 1.4, donde puede apreciarse la diferencia de la densidad de masa que se encuentra el rayo de luz dependiendo de si el sistema está alineado o no. Por otro lado está el *microlensing*, donde las masas de las lentes son menores y producen por lo tanto separaciones de imágenes menores que en el *strong lensing*. El nombre viene de hecho de estas separaciones, del orden de 10^{-6} arcosegundos.

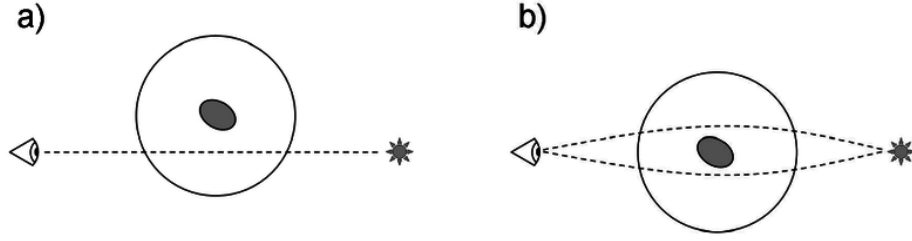


Figura 1.4. El diagrama muestra la diferencia entre el régimen débil (a) y el régimen fuerte (b). La lente está representada por una elipse sombreada y contenida en una circunferencia. El área sombreada es el centro de la lente, donde mayor es la densidad superficial de masa, y el resto tiene una densidad de masa menor, es decir, el campo gravitatorio en esta región es menos intenso. Por lo tanto, cuanto el eje óptico pasa por la región de densidad de masa superficial pequeña, el efecto es pequeño y nos encontramos en el régimen de *lensing* débil, mientras que en el caso del *strong lensing* los rayos de luz pasan por el centro de la lente y como el campo es intenso, se curvan mucho en su trayectoria. Imagen tomada del artículo *Gravitational Lensing as a Probe of Cold Dark Matter Subhalos*, de Teresa Riehm y Erik Zackrisson.

El desarrollo de este trabajo se centra en las lentes gravitacionales débiles, debido a que resulta ser un muy buen método de estimación de masas. En el régimen fuerte se observan anillos como los de las Fig. 1.2 y 1.3, por lo que a la hora de estimar la masa de la lente solo se tiene en cuenta la parte que está en el interior del radio de Einstein, dejando a un lado toda la masa que queda en las regiones externas del cúmulo, y este límite hace que la masa sea subestimada. El *weak lensing*, sin embargo, permite tener en cuenta separaciones angulares mayores porque, aunque débiles, las distorsiones ocupan un mayor rango de visión, como se observa en la Figura 1.5, y en lugar de considerar una sola distorsión grande se tendrá en cuenta el alineamiento general de las galaxias, pudiendo obtener las masas mediante análisis estadísticos. Además, el hecho de que en el régimen fuerte la lente necesite estar alineada con una fuente hace que este fenómeno sea más difícil de observar, mientras que el efecto débil se puede encontrar muy frecuentemente, pues todas las galaxias se ven afectadas en mayor o menor medida por la distribución de masa que encuentran los rayos de camino al observador.

1.1.1. Aplicaciones

Antes de continuar con nuestro objeto de investigación y con la derivación de las ecuaciones del *lensing*, parece importante destacar las distintas aplicaciones de una herramienta tan poderosa como son las lentes gravitacionales.

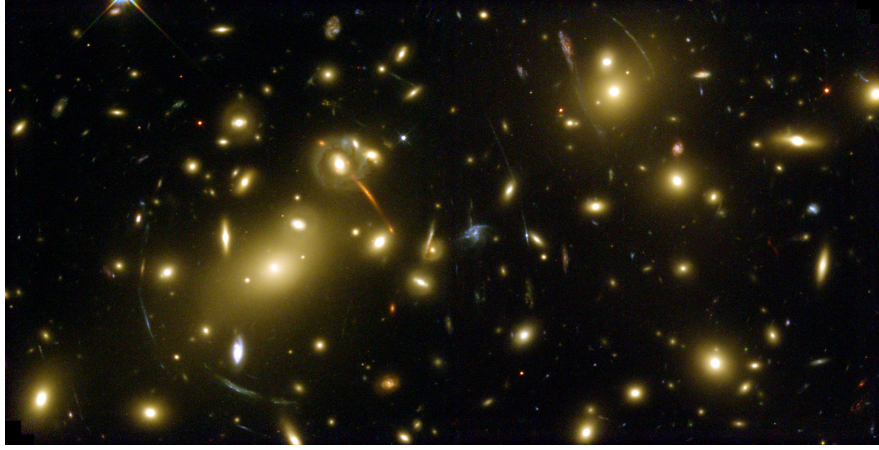


Figura 1.5. Lensado de un fondo de galaxias por el cúmulo de galaxias Abell 2218 en el régimen débil. Fuente: NASA / SCScl (<https://hubblesite.org/contents/media/images/2000/07/942-Image.html>).

Estimación de masas

Como ya se ha visto, la estimación de la distribución de masa de la lente es el principal objetivo de este trabajo. Como son los campos gravitatorios los que deflecan los rayos de luz y éstos se relacionan con la masa, el estudio de las lentes gravitatorias proporciona un buen método de estimación de masas de objetos astronómicos. En particular, una de las ventajas de este método es que el efecto de *lensing* gravitatorio es independiente del tipo de materia (ya sea luminosa o no), por lo que es igual de útil para materia oscura que para materia visible.

Parámetros cosmológicos

Otra de las aplicaciones más importantes de las lentes gravitacionales es la cosmografía. Como se ha mencionado anteriormente, algunos sistemas de lente pueden producir múltiples imágenes, donde rayos de luz de una misma fuente siguen trayectorias distintas. Cada rayo necesitará un tiempo diferente para llegar al observador, y por ello tienen un tiempo de retardo (*time delay*), inversamente proporcional a la constante de Hubble cosmológica H_0 [15]. También pueden estimarse otros parámetros cosmológicos, como los parámetros de densidad total de materia Ω_m y de densidad de energía oscura Ω_Λ [16].

Detección de fuentes muy lejanas

Debido a que las lentes pueden magnificar la fuente además de distorsionarla, sus imágenes pueden parecer más luminosas de lo que las veríamos sin lente. En particular hay ciertos objetos astronómicos que por ser lejanos o por tener poca luminosidad no somos capaces de detectar sin ayuda de una lente gravitatoria que los magnifique. En

este sentido, las lentes actúan como “telescopios naturales”.

1.2. El formalismo de lente

Tras esta breve introducción del fenómeno de *lensing* gravitatorio, a continuación se deducirán sus ecuaciones más fundamentales y en las que se basará este trabajo. Primero se derivará la ecuación de lente, que relaciona la posición real de la fuente con la posición observada a través de la lente, y después se introducirá el concepto de potencial de lente y otras ecuaciones destacables para las lentes gravitatorias. La derivación de estas ecuaciones sigue la notación y el desarrollo de Schneider, Kochanek y Wambsganss (2006) [17].

1.2.1. Ecuación de lente

La Fig. 1.6 muestra un esquema del sistema de lente gravitacional habitual; una distribución de masa (plano lente) deflexa los rayos provenientes de una fuente de luz (plano fuente), con un ángulo de deflexión $\hat{\alpha}$. Suponiendo distancias D_d y D_{ds} mucho mayores que la profundidad de la lente a lo largo del eje óptico (esto permite tomar la aproximación de ángulos pequeños), aunque las trayectorias de los rayos de luz sean curvadas aproximar a rectas es razonable. Como se ha mencionado, esta ecuación relaciona la posición real β de la fuente con la posición θ observada por el efecto de la lente.

Para trabajar desde el punto de referencia del observador se introduce el ángulo de deflexión escalado α , que siguiendo la Fig. 1.6 se relaciona por geometría y suponiendo ángulos pequeños con el ángulo de deflexión $\hat{\alpha}$ como:

$$\alpha(\xi) = \frac{D_{ds}}{D_s} \hat{\alpha}(\xi)$$

donde ξ es el parámetro de impacto señalado en la Sección 1.2, que puede relacionarse con las posiciones aparentes por

$$\xi = D_d \theta$$

De esta manera, se tiene que

$$\beta = \theta - \frac{D_{ds}}{D_s} \hat{\alpha}(D_d \theta) \equiv \theta - \alpha(\theta) \quad (1.1)$$

La ecuación de lente 1.1 puede generalizarse a espacios curvos si se utilizan distancias de diámetro angular, debido a la manera en que las definimos. Asimismo cabe destacar el carácter no-lineal de la ecuación de lente, lo que explica que para cada β pueda haber más de una solución de θ produciendo imágenes múltiples (en el régimen de *strong lensing*).

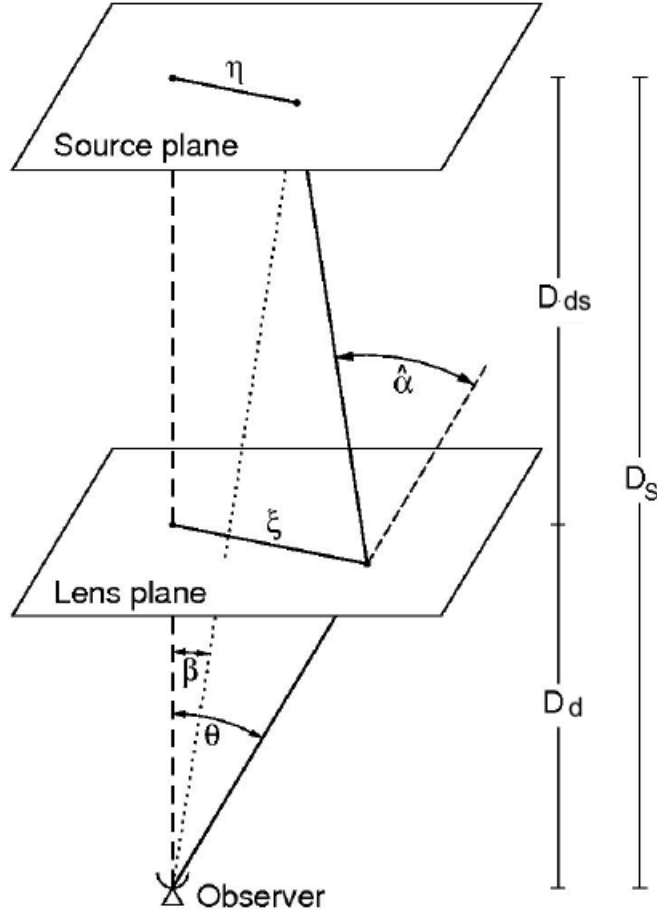


Figura 1.6. Esquema del sistema habitual de lente gravitatoria. Imagen tomada del libro de Schneider, Kochanek y Wambsganss (2006), *Gravitational Lensing: Strong, Weak and Micro*.

1.2.2. Ángulo de deflexión y potencial de *lensing*

Antes de introducir el concepto de potencial de lente conviene desarrollar la expresión del ángulo de deflexión en función de la densidad superficial de masa.

En la introducción histórica a las lentes se mencionó que, según las ecuaciones de la relatividad general, el ángulo de deflexión de un rayo de luz venía dado por

$$\hat{\alpha} = \frac{4GM}{c^2\xi}$$

donde M es una masa esférica y ξ el parámetro de impacto, siempre que ξ sea mucho mayor que el radio de Schwarzschild de la masa R_s , $\xi \gg R_s$. Esta condición implica que el ángulo de deflexión es pequeño, $\hat{\alpha} \ll 1$, y que nos encontramos en un régimen de campo gravitatorio débil. En este régimen las ecuaciones de campo de la Relativi-

dad General pueden ser linealizadas, y de esta forma el ángulo de deflexión $\hat{\alpha}$ de un conjunto de puntos de una masa será la suma vectorial de los $\hat{\alpha}$ causados por cada una de las componentes de masa individuales.

Supongamos una distribución volumétrica de masa $dm = \rho(\mathbf{r})dV$ y un rayo de luz que pasa por esta distribución con una trayectoria $(\xi_1(\lambda), \xi_2(\lambda), r_3(\lambda))$ de manera que el rayo incidente se propaga a lo largo de r_3 . Por la aproximación de lente delgada $\xi(\lambda) \approx \xi$, con $\xi = (\xi_1, \xi_2)$. Así, el vector de impacto del rayo de luz en el punto $\mathbf{r}' = (\xi'_1, \xi'_2, r'_3)$ es $\xi - \xi'$. Por lo tanto, el ángulo de deflexión total se desarrolla como:

$$\begin{aligned}\hat{\alpha}(\xi) &= \frac{4GM(\xi - \xi')}{c^2|\xi - \xi'|^2} = \frac{4G}{c^2} \sum dm(\xi'_1, \xi'_2, r'_3) \frac{\xi - \xi'}{|\xi - \xi'|^2} = \frac{4G}{c^2} \int \rho(\xi'_1, \xi'_2, r'_3) dV \frac{\xi - \xi'}{|\xi - \xi'|^2} \\ &= \frac{4G}{c^2} \int d^2\xi' \int dr'_3 \rho(\xi'_1, \xi'_2, r'_3) \frac{\xi - \xi'}{|\xi - \xi'|^2}\end{aligned}$$

Es importante notar que ξ , ξ' y $\hat{\alpha}(\xi)$ son vectores bidimensionales e independientes de r'_3 , de modo que se puede definir así la densidad superficial de masa $\Sigma(\xi)$:

$$\Sigma(\xi) \equiv \int dr_3 \rho(\xi_1, \xi_2, r_3) \quad (1.2)$$

siendo ésta la densidad de masa proyectada en el plano perpendicular al rayo de luz incidente.

Por lo tanto, el ángulo de deflexión causado por una densidad de masa viene dado por la siguiente expresión:

$$\hat{\alpha}(\xi) = \frac{4G}{c^2} \int d^2\xi' \Sigma(\xi') \frac{\xi - \xi'}{|\xi - \xi'|^2}$$

Al igual que para la ecuación de lente, interesa utilizar el ángulo de deflexión escalado para poder expresar esta cantidad en términos de la posición aparente θ :

$$\begin{aligned}\alpha(\theta) &= \frac{D_{ds}}{D_s} \hat{\alpha}(D_d \theta) = \frac{D_{ds}}{D_s} \frac{4G}{c^2} \int d^2(D_d \theta') \Sigma(D_d \theta') \frac{D_d(\theta - \theta')}{D_d^2 |\theta - \theta'|^2} \\ &= \frac{D_{ds} D_d}{D_s} \frac{4G}{c^2} \int d^2(\theta') \Sigma(D_d \theta') \frac{\theta - \theta'}{|\theta - \theta'|^2}\end{aligned}$$

A partir de aquí se define la densidad superficial de masa adimensional o convergencia $\kappa(\theta)$ como

$$\kappa(\theta) := \frac{\Sigma(D_d \theta)}{\Sigma_{cr}}$$

donde Σ_{cr} es la densidad superficial de masa crítica

$$\Sigma_{cr} = \frac{c^2}{4\pi G} \frac{D_s}{D_d D_{ds}}$$

La convergencia κ marca la posibilidad de que haya múltiples imágenes para algunas posiciones de la fuente, es decir, la densidad crítica es un valor característico que delimita los regímenes débil y fuerte del *lensing*. Para densidades por encima de este valor crítico entramos en el régimen fuerte, mientras que por debajo el ángulo de deflexión es más pequeño y estamos en un régimen débil.

Introduciendo la convergencia en la expresión del ángulo de deflexión se obtiene

$$\boldsymbol{\alpha}(\boldsymbol{\theta}) = \frac{1}{\pi} \int_{\mathbb{R}^2} d^2\theta' \kappa(\boldsymbol{\theta}') \frac{\boldsymbol{\theta} - \boldsymbol{\theta}'}{|\boldsymbol{\theta} - \boldsymbol{\theta}'|^2} \quad (1.3)$$

Llegados a este punto se puede introducir el concepto de potencial de *lensing* a partir de la Ec. 1.3, utilizando que $\nabla \ln |\boldsymbol{\theta}| = \boldsymbol{\theta}/|\boldsymbol{\theta}|^2$:

$$\varphi(\boldsymbol{\theta}) = \frac{1}{\pi} \int_{\mathbb{R}^2} d^2\theta' \kappa(\boldsymbol{\theta}') \ln |\boldsymbol{\theta} - \boldsymbol{\theta}'| \quad (1.4)$$

de manera que el ángulo de deflexión escalado queda escrito como el gradiente del potencial de lente (1.4) como:

$$\boldsymbol{\alpha} = \nabla \varphi$$

Del mismo modo, utilizando que $\nabla^2 \ln |\boldsymbol{\theta}| = 2\pi\delta_D(\boldsymbol{\theta})$ (δ_D es la función delta de Dirac bidimensional), se llega a la ecuación de Poisson escrita como:

$$\nabla^2 \varphi = 2\kappa \quad (1.5)$$

Finalmente se ha llegado a que la convergencia, y por consiguiente la densidad superficial de masa $\Sigma(\boldsymbol{\theta})$ puede relacionarse con el potencial de lente $\varphi(\boldsymbol{\theta})$ mediante la Ec. 1.5.

Capítulo 2

Weak lensing de cúmulos de galaxias sobre galaxias de fondo

La primera parte de este trabajo consiste en simular el lensado de un fondo de galaxias (plano fuente) por un cúmulo de galaxias (plano lente), con el fin de poder reconstruir la distribución de masa del cúmulo en el Capítulo 3 aplicando la maximización de la aleatoriedad del de-lensado. Para ello, siendo ésta una primera fase del estudio del método que se quiere implementar, se utilizarán simulaciones de imágenes simplificadas e idealizadas, por lo que se generan por un lado el modelo paramétrico de la lente gravitatoria y el fondo de galaxias aleatorio por otro, utilizando perfiles de densidad y de luminosidad adecuados, y se simula el *lensing* utilizando el paquete de *software* LensTools (Andrea Petri, 2014) [18], que calculará las trayectorias de los rayos de luz de la fuente al pasar por un plano de potencial de lente dado. En resumen, el código implementado deberá:

- Generar el potencial de *lensing* y la densidad superficial de masa del cúmulo de galaxias con un modelo paramétrico elíptico.
- Crear una imagen de galaxias distribuidas aleatoriamente.
- Lensar esta imagen de fondo con las funcionalidades de LensTools, utilizando el potencial de lente del cúmulo generado en el primer paso.

Las secciones que siguen describen la simulación por partes.

2.1. Simulación de la lente gravitatoria

Para simular los cúmulos de galaxias se ha utilizado un modelo elíptico con un perfil de densidad dado por la distribución Navarro-Frenk-White. Además de los planos de potencial de lente necesarios para el trazado de rayos, se han creado también mapas de densidad superficial de masa para poder dar cuenta de cómo se distribuye la masa, de manera que a cada cúmulo de galaxias modelizado por un conjunto de parámetros le correspondan un mapa de potencial y otro de densidad.

2.1.1. Perfil de masa de los cúmulos de galaxias: distribución de Navarro, Frenk y White

Mediante simulaciones cosmológicas de N-cuerpos de alta resolución, Navarro, Frenk y White (NFW) [19] observaron que cúmulos de galaxias con masas muy distintas seguían la misma distribución de densidad, llegando a la conclusión de que existe un perfil de densidad universal dado por la función:

$$\rho(r) = \frac{\delta_c \rho_c(z)}{(r/r_s)(1 + r/r_s)^2} \quad (2.1)$$

donde $\rho_c(z) = 3H(z)^2/(8\pi G)$ es la densidad crítica del Universo en z , δ_c es la densidad característica de cada cúmulo y r_s es el radio de escala.

La densidad característica de un cúmulo puede relacionarse con su parámetro de concentración c , definido como la razón entre el radio virial r_{200} (el radio en el interior del cual la densidad de masa media es $200\rho_c(z)$) y el radio de escala r_s , mediante la siguiente expresión [20]:

$$\delta_c = \frac{200}{3} \frac{c^3}{\ln(1+c) - c/(1+c)} \quad (2.2)$$

Aunque la función (2.1) diverge cuando $r \rightarrow 0$, diversas simulaciones numéricas muestran que el perfil de densidad de los cúmulos se aplanan en el centro [21, 22]. De todas formas, a pesar de que exista una singularidad central, se puede demostrar que el perfil de densidad (2.1) siempre tiene líneas críticas [23] y puede formar arcos radiales, por lo que en realidad no supone un problema, pero a la hora de realizar el cálculo numérico conviene utilizar una densidad constante en el centro del cúmulo para simplificar cálculos.

En nuestro código se toma el valor medio de los píxeles vecinos para el punto en que la función diverge.

2.1.2. Modelo de potencial elíptico NFW

Aunque los cúmulos de galaxias sean estructuras tridimensionales, su tamaño es muy pequeño en comparación con las distancias D_{ds} y D_s ¹ de las ecuaciones de lente, y además cuando se obtiene una imagen del cielo lo que se tiene es algo 2-dimensional, haciendo que sea necesario trabajar con la proyección del perfil NFW sobre el plano de la lente.

Por lo tanto, en este trabajo se proyectarán distribuciones de masa NFW sobre elipses. La elección de un modelo de lente elíptico se debe, además de a que resulte ser

¹Un cúmulo de galaxias típico tiene un diámetro del orden de 2 a 10 Mpc, en comparación con las distancias de $D_s \approx 1300$ Mpc y $D_{ds} \approx 1100$ Mpc con las que trabajaremos.

un modelo realista, a su simplicidad, ya que pueden obtenerse expresiones analíticas tanto para el potencial de lente elíptico $\varphi(r)$ como para su densidad superficial de masa $\Sigma(r)$ [24].

Para esta simulación se utilizarán expresiones existentes en la literatura [25]. Por comodidad se introduce la coordenada radial adimensional $x = r/r_s$. Integrando la distribución de densidad NFW (2.1) respecto al eje óptico z se obtiene la densidad superficial de masa $\Sigma(x)$, que toma la expresión

$$\Sigma(x) = \int_{-\infty}^{+\infty} \rho(r_s x, z) dz = 2\delta_c \rho_c r_s F(x) \quad (2.3)$$

con

$$F(x) = \begin{cases} \frac{1}{x^2-1} \left(1 - \frac{1}{\sqrt{1-x^2}} \operatorname{arccosh} \frac{1}{x} \right) & \text{si } x < 1 \\ \frac{1}{3} & \text{si } x = 1 \\ \frac{1}{x^2-1} \left(1 - \frac{1}{\sqrt{x^2-1}} \arccos \frac{1}{x} \right) & \text{si } x > 1 \end{cases}$$

Siguiendo la derivación de las ecuaciones de la sección 1.3.2, en concreto por la relación de la Ec. (1.5), el potencial de lente $\varphi(x)$ viene dado por

$$\varphi(x) = 2\kappa_s \theta_s^2 h(x) \quad (2.4)$$

con

$$h(x) = \begin{cases} \ln^2 \frac{x}{2} - \operatorname{arccosh}^2 \frac{1}{x} & \text{si } x < 1 \\ \ln^2 \frac{x}{2} + \arccos^2 \frac{1}{x} & \text{si } x \geq 1 \end{cases}$$

donde θ_s es el radio de escala angular y $\kappa_s = \delta_c \rho_c r_s \Sigma_{cr}^{-1}$ es la convergencia.

Para darle a estas expresiones contornos elípticos, la coordenada radial r deberá describir elipses, definidas por su centro (x_0, y_0) , su excentricidad ε y su orientación ϕ . Estos serán los parámetros de forma del modelo, a los que hay que añadir el parámetro de concentración c que caracteriza a cada cúmulo de galaxias, y que lleva a la masa total del cúmulo.

2.1.3. Simulaciones

Con lo previamente dicho, tenemos que los mapas de potencial (y correspondientemente los de densidad) podrán ser modelizados mediante seis parámetros: el centro de la lente (x_0, y_0) , el radio de escala r_s , la excentricidad de la elipse ε , su orientación ϕ y el parámetro de concentración c . La mayoría de cúmulos de galaxias tienen una excentricidad de entre 0.6 y 0.9 [26], lo que nos sirve para acotar los valores que vayamos a introducir en este parámetro. Por otra parte, podemos considerar que el cúmulo está

centrado en nuestra imagen, siempre con cierto error, suponiendo que cuando observamos el cielo podemos centrar esta lente. Los valores típicos de radios de escala [25, 20] rondan el intervalo de 0.05 a 0.30 Mpc, y el parámetro c puede tomar valores desde 5 hasta 40 [27]. Cabe señalar que se ha intentado asignar valores realistas a los parámetros para que éstos lleven a masas razonables y a cúmulos parecidos a los que existen en el Universo, pero siempre intentando mantener un efecto de lente gravitatoria suficientemente notorio como para poder apreciarlo en las Figuras, como primeros pasos a la hora de testar el algoritmo. Por ello, manteniendo los parámetros de forma realistas (éstos no contribuyen a la intensidad del potencial de *lensing*), se tomarán valores de $r_s \in [0,1, 0,5]$ para el radio de escala, algo mayores que los realistas, y $c \in [15, 25]$ para el parámetro de concentración, valores altos aunque entren en el rango de los valores típicos.

Teniendo en cuenta estas consideraciones, en las Fig. 2.1 y 2.2 vienen representados a modo de ejemplo los mapas de potencial y de densidad, respectivamente, en imágenes de 512x512 píxeles para los valores $(x_0, y_0) = (256, 256)$, $r_s = 0,50$ Mpc, $\varepsilon = 0,7$, $\phi = -1,0$ rad y $c = 25$. Nótese que se han tomado los valores más grandes del intervalo de r_s y de c para una mejor visualización del efecto de lente. Debido a que la mayor cantidad de masa está concentrada en el centro se ha representado también el logaritmo de la densidad.

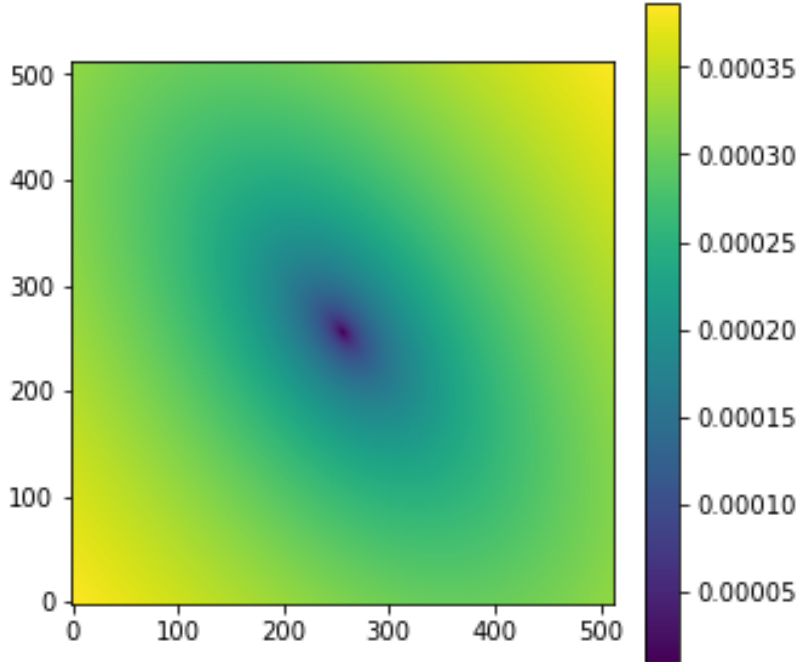


Figura 2.1. Mapa del potencial de lente de 512x512 píxeles para una lente elíptica con una distribución NFW, con parámetros $(x_0, y_0) = (256, 256)$, $r_s = 0,50$ Mpc, $\varepsilon = 0,7$, $\phi = -1,0$ rad y $c = 25$.

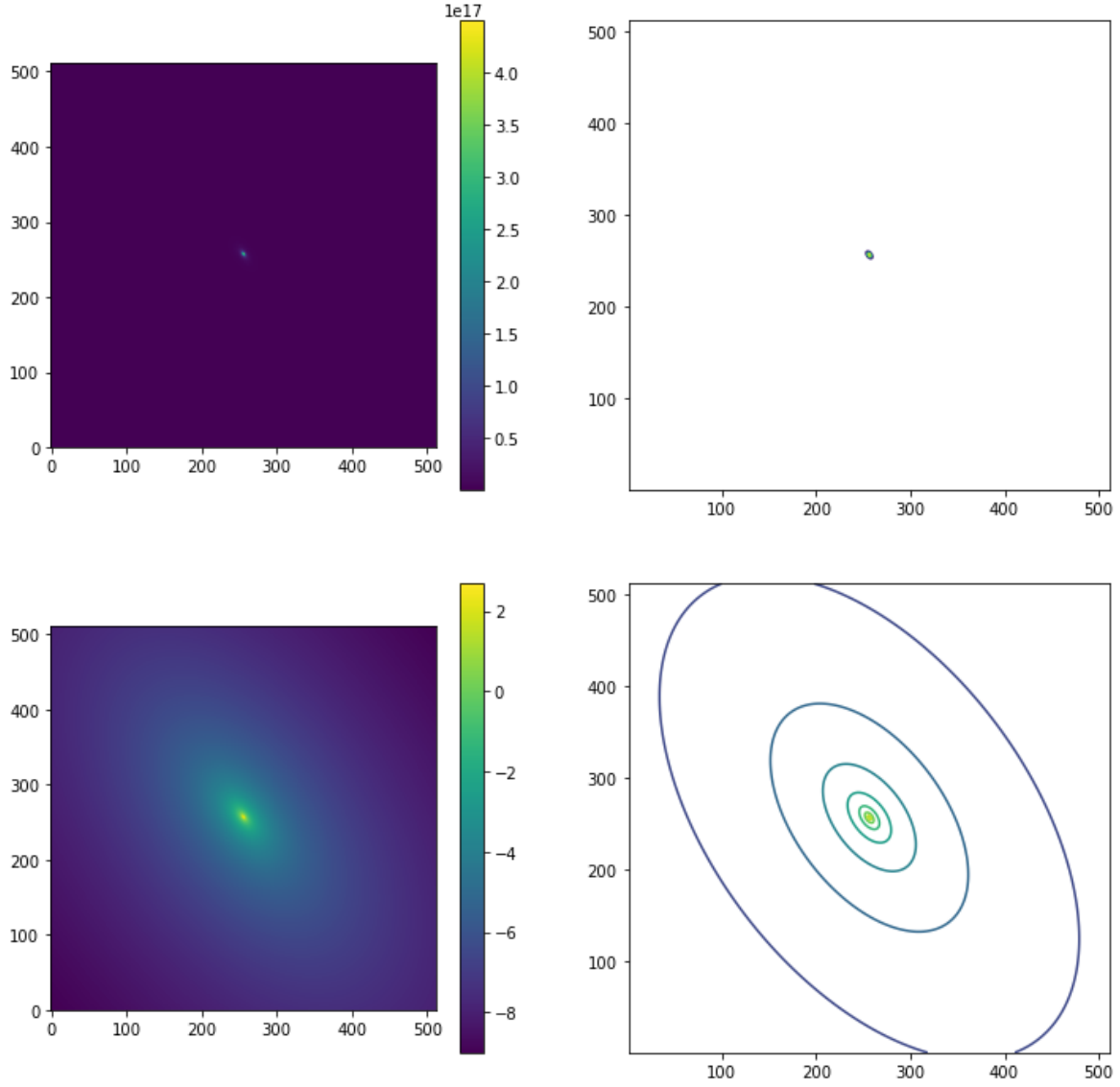


Figura 2.2. Mapas de la densidad superficial de masa NFW para un modelo de lente elíptico con parámetros $(x_0, y_0) = (256, 256)$, $r_s = 0,50$ Mpc, $\varepsilon = 0,7$, $\phi = -1,0$ rad y $c = 25$. Arriba, la densidad (*izq.*) y el mapa de contornos correspondiente (*dcha.*). Abajo, el logaritmo de la densidad (*izq.*) y su mapa de contornos correspondiente (*dcha.*).

En conclusión, el código implementado puede simular diversas lentes gravitatorias elípticas variando los seis parámetros mencionados y generar su potencial de lente $\varphi(r)$ y su densidad superficial de masa $\Sigma(r)$.

2.2. Simulación de galaxias

Para esta primera implementación del método, donde se utilizan simulaciones simplificadas e idealizadas como primer paso para una futura aplicación a imágenes más realistas, queda generar un fondo de galaxias a lensar. Manteniendo un modelo elíptico para generarlas, se distribuirán 120 elipses con parámetros (x_0, y_0) , ε y ϕ aleatorios en una imagen de 512x512 píxeles. A las galaxias se les ha proporcionado un perfil de luminosidad de De Vaucouleurs [28] dado por la siguiente expresión:

$$I(r)/I_0 = \exp \left[-7,66925[(r/r_e)^{1/4} - 1] \right]$$

con r_e el radio efectivo. A la intensidad I_0 también se le ha dado un valor aleatorio.

Aunque muchas investigaciones apuntan a que la luminosidad de las galaxias viene mejor descrita por un perfil de Sérsic generalizado a $r^{1/n}$ con $1/2 \leq n \leq 10$ [29], se ha optado por utilizar un perfil $r^{1/4}$ de De Vaucouleurs debido a que sigue dando una buena descripción de la dependencia de la luminosidad con el radio y dejar un valor fijo de n no afecta el desarrollo de nuestro trabajo, solo proporciona unas galaxias de fondo más uniformes.

En la Fig. 2.3 puede verse un ejemplo de una distribución aleatoria de galaxias, que se podrá usar como plano fuente para la simulación del *lensing* gravitatorio. De nuevo la imagen está representada en escala logarítmica, por razones meramente visuales.

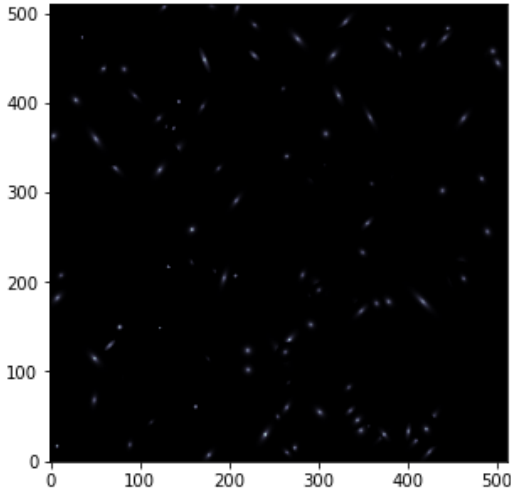


Figura 2.3. Generación aleatoria $((x_0, y_0)$, ε , ϕ e I_0 aleatorios) de galaxias con un perfil de luminosidad de De Vaucouleurs para el plano fuente de 512x512 píxeles, representación en escala logarítmica.

2.3. Lensado y de-lensado de galaxias

Para simular el lensado del fondo de galaxias (generado en la Sección 2.2) por un plano de lente elíptico (generado en la Sección 2.1) se ha empleado el paquete Lens-

Tools² desarrollado por Andrea Petri. Este código escrito en Python tiene diversas funcionalidades para el análisis de datos del *weak lensing*. En particular, se ha utilizado la simulación de trazado de rayos (*Ray Tracing Simulations*), donde hay que crear un sistema de lentes al que se añaden planos de potencial de lente, y el programa calcula entonces las trayectorias de los rayos de luz tomando un *array* de posiciones como argumento, tanto al lensar como al de-lensar, que más tarde habrá que convertir a imágenes que mantengan el perfil de luminosidad de las galaxias originales.

Además del plano de potencial, el código necesita el *redshift* z del plano fuente y del plano lente, junto con los parámetros cosmológicos H_0 , Ω_m y Ω_Λ y la apertura angular θ en grados. Los valores que se han fijado para esta simulación vienen recogidos en la Tabla 2.1. Los corrimientos al rojo z han sido fijados en función de un modelo realista con cúmulos relativamente cercanos, y los parámetros cosmológicos han sido obtenidos de la misión *Planck* 2018 [30].

La manera de pasar todos estos datos al *software* es mediante un archivo FITS, de manera que es necesario convertir el potencial de lente a este formato y añadir los valores de *redshift* z de la lente, los parámetros H_0 , Ω_m y Ω_Λ y la apertura θ al encabezado.

Parámetro	Valor fijado
z (lente)	0.1
z (fondo)	0.5
H_0	67.66 km s ⁻¹ Mpc ⁻¹
Ω_m	0.3111
Ω_Λ	0.6889
θ	3,5°

Tabla 2.1. Valores fijados para la simulación para cada uno de los parámetros utilizados por LensTools.

2.3.1. Resultados

Antes de calcular el *lensing* del sistema cúmulo-galaxias se va a calcular el trazado de rayos de una matriz de puntos (Fig. 2.4). Esto servirá de ayuda a la hora de entender y analizar el fondo de galaxias lensado. Para la lente creada en la Sección 2.1, con parámetros $(x_0, y_0) = (256, 256)$, $r_s = 0,50$ Mpc, $\varepsilon = 0,7$, $\phi = -1,0$ rad y $c = 25$, los efectos más notorios del *lensing* se aprecian en la esquina inferior derecha, así como en los bordes del *grid*. De hecho, observando la imagen nuevamente de-lensada puede verse que algunas posiciones de los bordes no recuperan su posición original al completo.

²Se puede encontrar la documentación en <https://lenstools.readthedocs.io/en/latest/#>

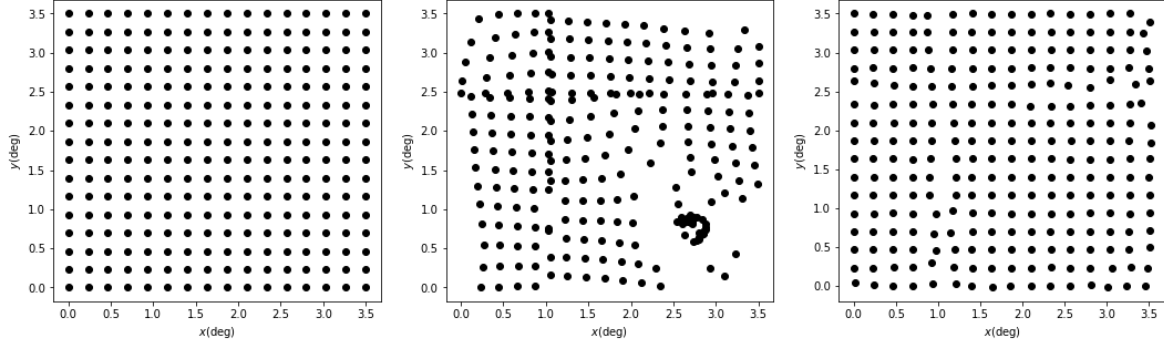


Figura 2.4. Lensado y de-lensado de un *grid* de puntos. En orden de izquierda a derecha, la matriz original, la matriz lensada y la matriz de-lensada.

Al lensar un fondo de galaxias cabe esperar un comportamiento muy similar al de la matriz de puntos. Si nos fijamos en la Fig. 2.5, la imagen lensada muestra un efecto de lensing considerable en la misma esquina inferior derecha de la Fig. 2.4, pero esta vez se observa una orientación de galaxias formando arcos radiales. Se pueden comparar esta imagen con las distorsiones por *weak lensing* mencionadas en el capítulo introductorio (Fig. 1.5).

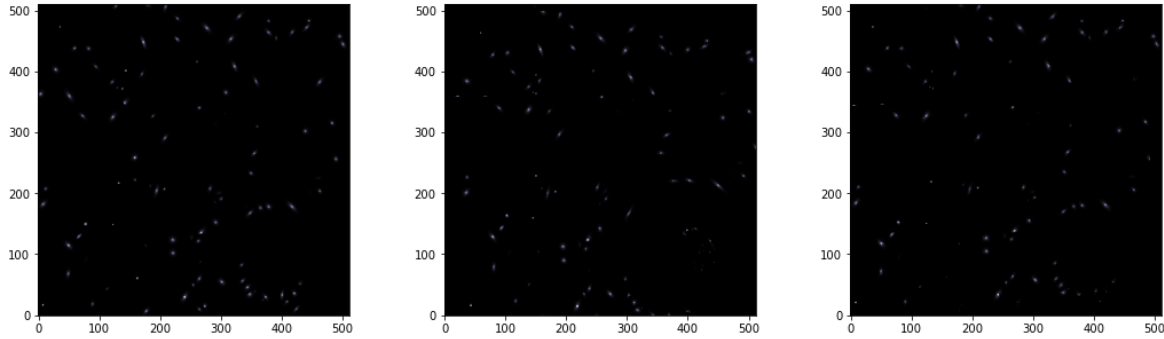


Figura 2.5. Lensado y de-lensado de un fondo de galaxias distribuidas aleatoriamente en imágenes de 512-512 píxeles. En orden de izquierda a derecha, la imagen original, la imagen lensada y la imagen de-lensada. Las imágenes están representadas en escala logarítmica.

Es importante notar que aunque las imágenes vuelvan a estar representadas en escala logarítmica, para calcular las trayectorias se han utilizado las distribuciones originales.

No obstante, el análisis del de-lensado de las galaxias no es tan rápida y visual como en el caso de los *grids*. Una manera de analizar en qué medida se recupera la imagen inicial con el de-lensado es hacer un estudio comparativo de la imagen original con la

imagen de-lensada. La Fig. 2.6 muestra la diferencia entre estas dos imágenes, con lo que se podrá ver cómo cambia la imagen con el lensado y el de-lensado. Observando esta figura puede apreciarse que existen diferencias debidas a un desplazamiento más o menos uniforme de las galaxias, sobre todo en la zona en la que más se notaba la distorsión del *lensing*, pero la imagen inicial se recupera con la misma distribución. Esto permite confirmar que al de-lensar imágenes lensadas estaremos en efecto volviendo a imágenes más o menos originales.

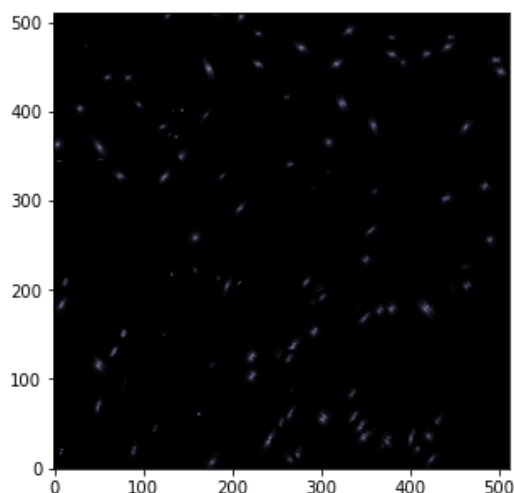


Figura 2.6. Logaritmo de la diferencia entre el fondo de galaxias original y el fondo de galaxias tras lensarlo y de-lensarlo.

Capítulo 3

Estimación de la lente a través de la maximización de la aleatoriedad

A pesar del amplio abanico de investigaciones existentes en el campo de las lentes gravitatorias, no se ha encontrado ningún estudio previo que aplique la maximización de la aleatoriedad de las imágenes de-lensadas como técnica de estimación de masas.

Tras haber generado las distribuciones de masa de lentes y las galaxias aleatorias y haber hecho unas pruebas de *lensing* gravitatorio, en este capítulo se aborda el método de maximización de la aleatoriedad estudiando su viabilidad en una primera aproximación, y se proponen posibles enfoques para continuar su desarrollo.

3.1. Estudio de la aleatoriedad

Antes de proceder con la maximización de la entropía se realizará una pequeña introducción teórica al concepto de entropía de Shannon y un análisis estadístico sobre la aleatoriedad de los fondos de galaxias, para estudiar en qué medida es factible reconstruir la distribución de masa de un cúmulo de galaxias por el método de máxima aleatoriedad.

3.1.1. Entropía global de Shannon

En teoría de la información, la entropía de Shannon mide la aleatoriedad de una cantidad de información. Esta entropía, denotada por $H(X)$, viene dada por [31]:

$$H(X) = - \sum_{l=1}^{L-1} P_l \log_2 P_l$$

donde X es una variable aleatoria y P_l es la probabilidad $Pr(X = l)$. La base del logaritmo nos da las unidades de la entropía, en base 2 las unidades son *bits* o *shannons*.

Otras bases habituales incluirían el logaritmo decimal y el logaritmo neperiano.

En el contexto de una imagen, si $l \in \{0, 1, \dots, L-1\}$ es el valor de la intensidad o del color de cada píxel X perteneciente a una matriz $M \times N$, P_l será la probabilidad de que un píxel tome el valor l y la entropía de Shannon de la imagen medirá la aleatoriedad de las intensidades de sus píxeles [32].

Resultados

Para calcular la entropía de Shannon de las imágenes generadas se ha utilizado una rutina de la librería Scikit-image [33]. El objetivo de este pequeño estudio es verificar que la aleatoriedad disminuye al lensar un fondo de galaxias, por lo que se han generado 4000 imágenes aleatorias de galaxias y se ha calculado la entropía de las imágenes originales y de las imágenes lensadas con la lente creada anteriormente con parámetros $(x_0, y_0) = (256, 256)$, $r_s = 0,50$ Mpc, $\varepsilon = 0,7$, $\phi = -1,0$ rad y $c = 25$. Se han creado histogramas con los valores de aleatoriedad obtenidos, que se han ilustrado en la Fig. 3.1. La Tabla 3.1 recoge los valores medios con sus correspondientes desviaciones.

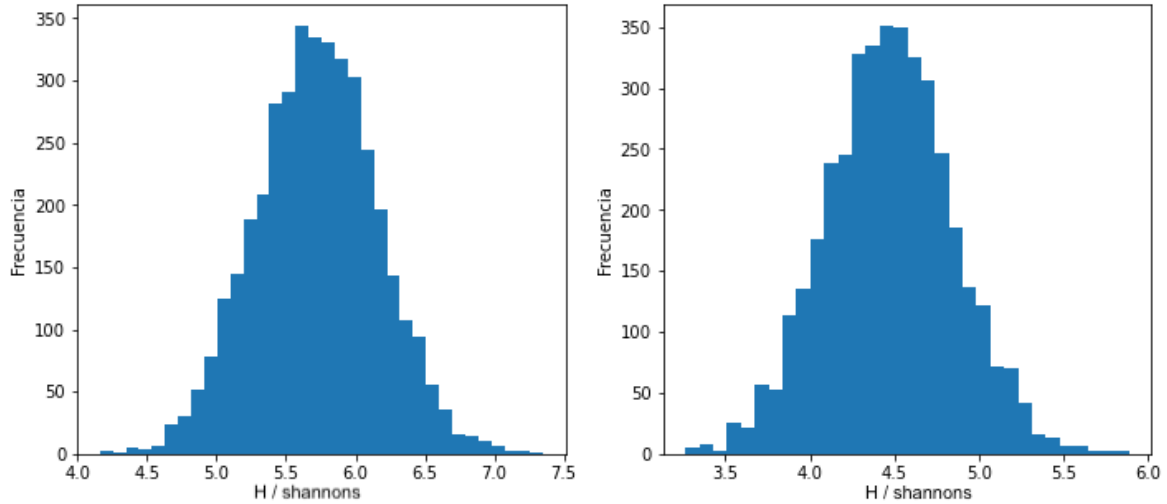


Figura 3.1. Histogramas de la entropía global de Shannon para 4000 muestras aleatorias de fondos de galaxia. A la izquierda la distribución de entropías de las imágenes originales, y a la derecha la distribución de entropías de las imágenes lensadas.

Imagen	Valor medio μ_H	Desviación estándar σ_H
Galaxias originales	5.7	0.4
Galaxias lensadas	4.5	0.4

Tabla 3.1. Estadística de la entropía de las imágenes original y lensada para una muestra de 4000 imágenes.

Si se observa los valores de la Tabla 3.1, es interesante que la diferencia entre los valores medios μ_H obtenidos para las galaxias originales y las lensadas es significativamente mayor que las desviaciones estándar σ_H . Esto nos permite afirmar que, en conjunto, estos resultados indican una tendencia a la disminución de la aleatoriedad cuando se lensa una imagen, lo que sugiere que maximizar la aleatoriedad de una imagen de-lensada para estimar el potencial de la lente que lo ha lensado es razonable.

3.2. Maximización de la aleatoriedad: determinación de la distribución de masa

Finalmente, pasamos a aplicar la maximización de la aleatoriedad como técnica de estimación de masas, para lo cual se utilizará el siguiente algoritmo:

- Paso 1. Generar un fondo de galaxias aleatorio.
- Paso 2. Lensar el fondo con un cúmulo de galaxias de parámetros (x_0, y_0) , r_s , ε , ϕ y c conocidos.
- Paso 3. De-lensar la imagen lensada con distintos potenciales, variando los parámetros (x_0, y_0) , r_s , ε , ϕ y c , y calcular la entropía de Shannon de cada nueva imagen de-lensada.
- Paso 4. Obtener los parámetros que generan la máxima aleatoriedad en el paso anterior.

Una vez obtenidos estos parámetros estimados pueden compararse con los valores reales, utilizados en el Paso 2. Finalmente, con los parámetros obtenidos ya puede generarse una reconstrucción de la distribución de masa del cúmulo de galaxias, utilizando para ello un mapa de densidad superficial de masa $\Sigma(r)$, y estimar así la masa del cúmulo.

El código con la implementación del algoritmo se adjunta en el Anexo A junto con las demás funciones creadas para el desarrollo del trabajo.

Veamos a modo de ejemplo la estimación de la densidad superficial de masa $\Sigma(r)$ del cúmulo de galaxias generado en la Sección 2.1.3, con parámetros $(x_0, y_0) = (256, 256)$, $r_s = 0,50$ Mpc, $\varepsilon = 0,7$, $\phi = -1,0$ rad y $c = 25$. El *lensing* es el mismo que el ilustrado en la Fig. 2.5, pero a la hora de de-lensar utilizaremos la rutina de optimización de funciones de la librería SciPy [34], que haciendo un barrido de todos los parámetros identificará los parámetros que maximizan la entropía de las imágenes de-lensadas. Para implementar esta maximización, la rutina de optimización utilizará el algoritmo de región de confianza para problemas con restricciones (*trust-constraint*), que aproxima la función a optimizar por una función modelo en una región considerada de confianza. Se ha optado por este método debido a que es el que mejor precisión proporciona.

dentro de los algoritmos existentes para problemas con restricciones. En realidad, esta rutina implementa la minimización de la función, por lo que en lugar de maximizar la función $H(X)$ se ha tenido que minimizar la función $-H(X)$.

Implementando este método se obtienen los parámetros recogidos en la Tabla 3.2, donde puede apreciarse que el valor del parámetro de concentración c es el que más se aleja de su valor real, pero si miramos la desviación relativa al valor real, los parámetros con más error son el radio de escala r_s y la orientación ϕ .

Parámetro	Valores reales p	Valores estimados \hat{p}	$\Delta p = p - \hat{p}$
x_0	256.0000	256.0000	0.0000
y_0	256.0000	256.0000	0.0000
r_s	0.5000	0.2840	0.2160
ε	0.7000	0.7476	-0.0476
ϕ	-1.0000	-0.7485	0.2515
c	25.0000	24.5097	0.4903

Tabla 3.2. Valores reales y valores estimados de los parámetros (x_0, y_0) , r_s , ε , ϕ y c de un cúmulo de galaxia, utilizando el método de maximización de aleatoriedad, junto con la diferencia entre el valor real y el estimado.

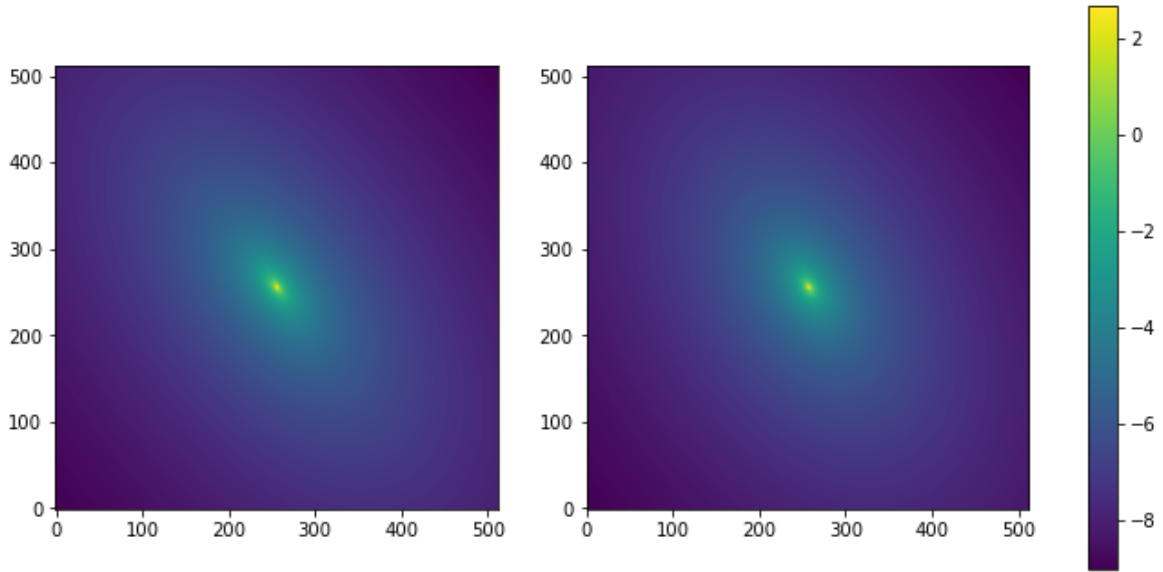


Figura 3.2. Distribuciones de la densidad superficial de masa $\Sigma(r)$ del cúmulo de galaxias real (*izq.*) y del cúmulo de galaxias reconstruido (*dcha.*), ambos en escala logarítmica.

En la Fig. 3.2 vienen dibujados los mapas de densidad superficial de masa $\Sigma(r)$ de los cúmulos real y estimado. A primera vista las dos distribuciones son muy simi-

lares, lo que nos da una idea de cómo de buena es la reconstrucción. La densidad ha vuelto a representarse en escala logarítmica para poder visualizar mejor la distribución.

No obstante, para poder discutir con detalle la medida en que se aproximan los valores estimados a los reales, se debe hacer un estudio exhaustivo para diferentes cúmulos de galaxias. Para ello se repetirán los pasos del ejemplo anterior para lentes con distintos parámetros. A la hora de recorrer estos parámetros se tendrán en cuenta ciertas restricciones. Como se ha mencionado anteriormente, se considerará que el cúmulo está más o menos centrado en la imagen, por lo que recorreremos este parámetro en un intervalo pequeño, (255, 257) para x_0 y y_0 , y se crearán cúmulos realistas con excentricidades ε en un intervalo (0.6, 0.9). La orientación ϕ irá de $-\pi/2$ a $\pi/2$, el radio de escala r_s de 0.1 a 0.5 Mpc y el parámetro de concentración c de 15 a 25. Estos intervalos se han elegido en función de las consideraciones descritas en la Sección 2.1.3. Haciendo un barrido por todos estos parámetros, se han obtenido pares de parámetros reales frente a estimados para cada cúmulo distinto.

La mejor manera de analizar los resultados obtenidos parece ser calculando la diferencia entre parámetro real y estimado de todos los cúmulos, $\Delta p = p - \hat{p}$, siendo p cada uno de los seis parámetros, y determinando el valor medio y la desviación estándar de estas diferencias para cada parámetro. La situación más preferible sería tener diferencias nulas de media entre los valores reales y estimados, lo que significaría que en conjunto el método aproxima bien la distribución del cúmulo inicial, con desviaciones pequeñas. Los resultados estadísticos de las diferencias se han recogido en la Tabla 3.3, con su valor medio μ y su desviación estándar σ .

Parámetro	Valor medio $\mu_{\Delta p}$	Desviación estándar $\sigma_{\Delta p}$
Δx_0	1.07e-15	0.12
Δy_0	1.07e-15	0.12
Δr_s	1.18e-17	0.10
$\Delta \varepsilon$	-4.93e-19	0.10
$\Delta \phi$	-0.018	0.26
Δc	-0.039	0.16

Tabla 3.3. Valor medio de la diferencia Δp junto con su desviación estándar para cada parámetro.

Se observa que los valores que mejor se aproximan son las coordenadas del centro del cúmulo, (x_0, y_0) , con una diferencia media nula entre los parámetros real y estimado y con una desviación pequeña en relación con los valores reales, que entraban en el intervalo $x_0, y_0 \in [255, 257]$. La estimación del radio de escala r_s y la excentricidad ε es también bastante rigurosa, el valor medio de la diferencia es nulo en ambos casos, pero esta vez, aunque la desviación estándar sea menor que para el centro de la distribución, esta variación tiene mayor importancia relativa, debido a que los valores reales están en intervalos $[0.1, 0.5]$ y $[0.6, 0.9]$. Por otro lado, la orientación ϕ también tiene

una diferencia media muy cercana al 0, entrando dentro del margen de error contando con la desviación, pero esta estimación es menor precisa que para los parámetros del centro del cúmulo por la misma razón que en el caso de la excentricidad y el radio de escala, el error relativo es mayor por tener valores reales pequeños, de $\pi/2$ a $\pi/2$. Por último, el parámetro de concentración c tiene un valor muy cercano al cero, aunque sea el que más se aleja, pero esta vez la desviación es muy pequeña comparada con los valores reales que puede tomar el parámetro.

Para visualizar mejor las diferencias y obtener la correlación entre los parámetros podrían dibujarse unos histogramas, pero el muestreo ha sido muy limitado para ello y no se obtiene una tendencia clara de cómo se distribuyen los errores. Esto nos permitiría dibujar un diagrama de correlación triangular con las matrices de correlación donde se viera la medida en que el error en un parámetro depende de los demás. Para estudiar mejor este aspecto, habría de aumentarse el número de iteraciones. No obstante, y debido al orden de complejidad algorítmica del código (para hacer tantas iteraciones con seis parámetros hace falta seis bucles `for` uno dentro del otro), para hacer tantas iteraciones se requiere que el ordenador esté ejecutando durante muchos días, por lo que esto se deja para un futuro estudio de la precisión.

En general la reconstrucción de la distribución es considerablemente precisa. También se puede ver que aunque las estimaciones del ejemplo de la Tabla 3.2 tuvieran diferencias mayores que las que se calculan después para una muestra grande de lentes, en la representación de la Fig. 3.2 no se puede apreciar mucha diferencia entre la distribución real y la estimada, por lo que las estimaciones en conjunto serán más similares aún.

3.3. Estimación de la masa de un cúmulo

Una vez obtenidos los parámetros para la reconstrucción de la distribución de masa, se puede calcular la masa total de un cúmulo dentro de un cierto radio R_{max} integrando en r . Si se tuviera un modelo esférico, la integral vendría dada por la expresión [35]

$$M = \int_0^{R_{max}} 4\pi r^2 \rho(r) dr = 4\pi \delta_c \rho_c r_s^3 \left[\ln \left(\frac{r_s + R_{max}}{r_s} \right) + \frac{r_s}{r_s + R_{max}} - 1 \right] \quad (3.1)$$

donde $\rho(r)$ es la distribución de NFW dada por la Ec. (2.1).

Para el caso de un modelo elíptico esta integral es más complicada de resolver, pero puede expresarse la masa total M de la misma forma:

$$M = \int_0^{R_{max}} 4\pi r^2 \rho(r) dr = \int_0^{R_{max}} 4\pi r^2 \frac{\delta_c \rho_c(z)}{(r/r_s)(1 + r/r_s)^2} dr$$

donde δ_c es la densidad característica de cada cúmulo, y se relaciona con el parámetro de concentración c por la Ec. (2.2). De esta forma, se tendría la masa total

$$M = \int_0^{R_{max}} 4\pi r^2 \frac{200}{3} \frac{c^3}{\ln(1+c) - c/(1+c)} \frac{\rho_c(z)}{(r/r_s)(1+r/r_s)^2} dr$$

en función de los parámetros de radio de escala r_s y de concentración c , obtenidos mediante el método de maximización de aleatoriedad.

Para comprobar en qué medida se obtienen masas realistas con los valores de los parámetros r_s y c utilizados en este trabajo, puede hacerse una estimación aproximada utilizando la fórmula para el modelo esférico, aunque se trabaje en realidad con un modelo elíptico. Tomemos los valores más grandes dentro de los intervalos de r_s y c , $r_s = 0,5$ Mpc y $c = 25$, y los más pequeños, $r_s = 0,1$ Mpc y $c = 15$. Con estos dos pares de valores, aplicando la Ec. (3.1) para la masa dentro de un radio $R_{max} = 4$ Mpc, teniendo en cuenta que los cúmulos de galaxia suelen tener diámetros de entre 2 y 10 Mpc, se obtienen las masas $M = 1,2 \cdot 10^{17} M_\odot$ para el cúmulo más grande y $M = 5,5 \cdot 10^{14} M_\odot$ para el cúmulo más pequeño. Las masas típicas de los cúmulos de galaxia varían en el rango de $10^{14} M_\odot$ a $10^{15} M_\odot$ [36], por lo que podemos afirmar que los parámetros utilizados han sido, aunque no muy alejados de la realidad, más altos de lo que deberían. Esto nos ha servido para visualizar mejor el efecto de lente que producen los cúmulos en esta primera fase del estudio del método de maximización de la aleatoriedad.

Es interesante comparar la masa $M = 1,2 \cdot 10^{17} M_\odot$ obtenida para los parámetros dados $r_s = 0,5$ Mpc y $c = 25$, que es el caso de la lente con la que se ha trabajado como ejemplo, con la masa para los valores estimados de $r_s = 0,5$ Mpc y $c = 25$, dentro de un mismo radio $R_{max} = 4$ Mpc, parámetros con los cuales se obtiene una masa total de $M = 2,8 \cdot 10^{16} M_\odot$. La masa estimada es menor que la real, porque ambos parámetros estimados quedaban por debajo de los valores reales, pero en un estudio más exhaustivo donde se ha estimado una mayor muestra, la diferencia entre los parámetros real y estimado se anulaba de media, por lo que las masas también se igualarán en conjunto.

Cabe recordar que estas estimaciones han sido realizadas con un modelo esférico, mientras que el modelo utilizado a lo largo del trabajo ha sido elíptico, pero ha servido de aproximación para la determinación de la masa de la lente.

3.4. Discusión del método y futuras investigaciones

En este último capítulo se ha comprobado que aplicando el método de maximización de la aleatoriedad se puede reconstruir la distribución de masa de cúmulos de galaxias con bastante precisión, obteniendo su densidad superficial de masa $\Sigma(r)$, y determinar la masa total del cúmulo a partir de los parámetros estimados obteniendo cúmulos que

se asemejan a los utilizados como lente. Además, las masas obtenidas son del orden de los cúmulos de galaxias reales.

No obstante, se debe tener precaución al interpretar los resultados obtenidos, pues conllevan cierto sesgo. El método ha sido aplicado sobre un modelo paramétrico simplificado, pero no se puede asegurar la fiabilidad de la técnica de reconstrucción para modelos más realistas, como uno que incluya una lente modelizada por varias elipses o lentes descentradas, por ejemplo. El problema de los modelos paramétricos es que están limitados a lentes que puedan ajustarse bien a ellos, pues pueden no existir expresiones analíticas del potencial de lente y de la densidad superficial de masa, pero normalmente pueden llegar a modelizarse objetos muy realistas superponiendo elipses, de manera que la aplicación de las ecuaciones de NFW elípticas (2.3) y (2.4) siga siendo viable.

Además, partimos de que lensar un fondo aleatorio ordena las galaxias, pero hay casos en los que esto no puede asegurarse. Tomemos el ejemplo de un fondo generado aleatoriamente, donde las galaxias están ordenadas muy uniformemente (en una generación aleatoria, este caso es una posibilidad). Si en la imagen original la distribución de galaxias es completamente uniforme y ordenada, es inevitable que lensarlas lleve a una imagen más aleatoria. Si bien hemos visto al principio del Capítulo 3 que generalmente la aleatoriedad disminuye con el *lensing*, la cantidad de muestras era relativamente pequeña comparada con las infinitas distribuciones aleatorias permitidas, y además la maximización de la entropía solamente ha sido estudiada para un fondo de galaxias determinado. En un estudio más exhaustivo se debería investigar la reconstrucción de masas para distintos fondos de galaxia.

En cuanto a la precisión del método, observando los resultados sobre la medida en que los parámetros estimados se acercan a los valores reales puede surgir la pregunta de cómo influye el error de uno de los parámetros sobre todos los demás. ¿Tener mucha precisión en uno de los parámetros implica necesariamente un desajuste en algún otro? Convendría estudiar la correlación entre los errores de los parámetros mediante diagramas triangulares de correlación. Para esto será necesario aumentar el muestreo en gran medida. Al mismo tiempo, al implementar el algoritmo de maximización, la rutina parte de ciertos intervalos donde buscar cada parámetro, se podría analizar qué pasaría si los intervalos dados fueran más grandes, suponiendo que no se conocen bien los parámetros del cúmulo real.

En lo concerniente a las medidas de aleatoriedad, es importante tener en cuenta que se ha utilizado una entropía global de Shannon. Este tipo de entropía no siempre se ajusta bien a la definición intuitiva de aleatoriedad, debido a que no tiene en cuenta la correlación entre dos píxeles, es decir, sus agrupamientos en estructuras locales, sino solamente las probabilidades globales. Una manera más apropiada de medir la aleatoriedad sería utilizando una medida de la entropía local, es decir, que tuviera en cuenta la distribución de aleatoriedades a lo largo de diferentes trozos de la imagen, como

proponen Y. Wu *et al.* [32].

En un desarrollo más exhaustivo de este método, sería interesante estudiar el lensado por varias lentes, suponiendo varios planos de potencial a distintos *redshifts* z . De la misma forma, podrían utilizarse como fondo a lensar varios planos de galaxias, cada una con un *redshift* diferente. Aunque el trabajo sólo cuente con fondos de galaxia simulados, también se podrían utilizar imágenes reales y lensarlas manteniendo y expandiendo la complejidad del modelo paramétrico de las lentes.

En conclusión, es necesario considerar una serie de limitaciones importantes que tiene esta técnica de reconstrucción de masas, pero en conjunto tras obtener estos resultados tan prometedores el método parece ser apropiado y su desarrollo requerirá un estudio más detallado.

Capítulo 4

Conclusiones

Siendo la materia oscura uno de los principales constituyentes del Universo, estudiarla se ha convertido en uno de los retos más importantes de la comunidad científica en la actualidad. Sin embargo, al no ser luminosa, la única manera de detectar la materia oscura es mediante sus efectos gravitatorios, y es aquí donde las lentes gravitacionales desempeñan un papel fundamental, convirtiéndose en una herramienta muy poderosa para el estudio del Universo.

Con el principal objetivo de explorar una nueva técnica de reconstrucción de masas basada en el efecto de lente gravitatoria y en un algoritmo de maximización de aleatoriedad de imágenes, en el desarrollo de este trabajo se ha estimado la distribución de masa de un cúmulo de galaxias y se ha estudiado la viabilidad de este método en una primera aproximación. Esto, teniendo en cuenta que la materia oscura constituye el 90 % de la masa de los cúmulos de galaxias, supone a su vez una herramienta para entender la materia oscura y una vía de estudio de la formación y evolución de la estructura a gran escala del Universo.

Para reconstruir la masa de un cúmulo de galaxias que actúa como lente se ha maximizado la aleatoriedad de las imágenes de-lensadas de unas galaxias de fondo, utilizando la entropía de Shannon para cuantificar dicha aleatoriedad. Debido a que ésta ha sido una fase preliminar del proceso de estudio del método, se ha optado por utilizar una metodología en la que la complejidad de aplicación de la técnica aumenta gradualmente. Por ello, en este trabajo se ha implementado un código que genera, por un lado, perfiles de densidad de masa (y, consecuentemente, potenciales de lente) que simulan a primer orden los perfiles que tendrían cúmulos de galaxias muy simplificados utilizando un modelo paramétrico elíptico y un perfil de masa dado por la distribución de NFW, y por otro lado fondos de galaxias aleatorias. El cúmulo vendrá descrito por seis parámetros: el centro del cúmulo (x_0, y_0) , el radio de escala r_s , la excentricidad ε , la orientación ϕ y el parámetro de concentración c . Tras lensar las galaxias con el cúmulo utilizando el paquete de *software* LensTools, se obtienen los parámetros de la lente que maximizan la aleatoriedad de las galaxias de-lensadas, y se comparan estos

parámetros con los parámetros de la lente real. A partir de dos de estos parámetros, el radio de escala y el parámetro de concentración, se puede calcular la masa total del cúmulo de galaxias, y compararla con cúmulos realistas.

A lo largo del trabajo se ha utilizado el ejemplo de una lente con parámetros $(x_0, y_0) = (256, 256)$, $r_s = 0,50$ Mpc, $\varepsilon = 0,7$, $\phi = -1,0$ rad y $c = 25$, ilustrando así la reconstrucción de masa para un cúmulo dado, antes de estudiar una muestra más amplia de lentes. Con los mapas de densidad de masa y de potencial de la lente generados, se ha podido estudiar el lensado de galaxias de fondo a través de las funcionalidades de LensTools, en particular la simulación del trazado de rayos, utilizando rutinas a las que se le pasan como argumento los mapas de potencial de lente en archivos FITS. Esto nos ha permitido obtener una visualización clara del efecto *lensing* para el perfil de masa dado. También a modo de estudio previo, se ha medido la entropía de una muestra de imágenes originales y lensadas para verificar que lensar un fondo de galaxias provoca que se alineen y disminuye por tanto su aleatoriedad. El código generado en esta primera parte ha servido para la aplicación del algoritmo de maximización de aleatoriedad, donde se ha creado una función que mide la entropía de imágenes de-lensadas pasándole los parámetros de una lente y una imagen lensada como argumentos. Es esta función la que se ha maximizado utilizando una rutina de optimización externa, obteniendo así una reconstrucción de la lente inicial.

Esta implementación se ha repetido para una mayor muestra de lentes, de manera que se ha podido estudiar la precisión de la estimación de los parámetros. Teniendo en cuenta los valores medios de las desviaciones, en conjunto se recupera el valor del parámetro x_0 con una precisión de $(1,07 \cdot 10^{-15} \pm 0,12)$, el parámetro y_0 con una precisión de $(1,07 \cdot 10^{-15} \pm 0,12)$, el valor de r_s con precisión $(1,18 \cdot 10^{-17} \pm 0,10)$, el valor de la excentricidad ε con un error de $(-4,93 \cdot 10^{-19} \pm 0,10)$, el parámetro ϕ con precisión $(-0,018 \pm 0,26)$ y el valor del parámetro de concentración c con una precisión de $(-0,039 \pm 0,16)$. Se aprecia de esta forma que los resultados son prometedores, los parámetros estimados se acercan razonablemente a los valores reales y por lo tanto el método de maximización de la aleatoriedad parece apropiado para conseguir una reconstrucción de la masa de una lente gravitatoria.

Sin embargo, éste no ha sido más que un estudio preliminar de un programa mucho más ambicioso, por lo que la importancia de los resultados obtenidos es que sirven para demostrar que la idea es factible y que merece la pena continuar con un estudio más detallado del método. La implementación de esta técnica todavía presenta muchas limitaciones que deberán resolverse en futuras fases de su desarrollo, como ampliar el muestreo para el estudio de la precisión de las reconstrucciones o aplicar el método a lentes más realistas y complejas.

En conclusión, explorar una nueva técnica para reconstruir la masa de cúmulos de galaxias en un campo de investigación tan relevante para la Astrofísica moderna

como son las lentes gravitatorias ha supuesto, en general, adentrarse y profundizar en los conocimientos sobre Física, y en particular aprender un lenguaje de programación ampliamente utilizado en la astrofísica, desenvolviéndose en el uso de herramientas externas (como por ejemplo, LensTools) y desarrollando un código propio, obteniendo así, junto con el estudio de la entropía, un enfoque multidisciplinar.

Apéndice A

Código

A continuación se presenta el código desarrollado para la implementación del método de maximización de aleatoriedad con el objetivo de reconstruir la distribución de masa de un cúmulo de galaxias.

Para su desarrollo, además de utilizar el *software* LensTools [18], se ha hecho uso de diversas librerías como son NumPy [37], matplotlib [38], astroPy [39, 40], skimage [33] y SciPy [34].

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3 """
4 Método de maximización de la aleatoriedad para la reconstrucción
5 de cúmulos de galaxias.
6
7 Código que genera cúmulos de galaxias elípticos con perfil de masa
8 NFW y galaxias de fondo aleatorias con perfil de luminosidad de
9 De Vaucouleurs, obtiene las imágenes del fondo lensado y de-lensado
10 utilizando la librería LensTools, estudia la entropía de Shannon de
11 las imágenes con y sin efecto de lensing e implementa un método de
12 maximización de la aleatoriedad de las imágenes
13 de-lensadas, obteniendo así unos parámetros estimados del cúmulo
14 inicial.
15 Se estudia la precisión del método estimando los parámetros de una
16 muestra de lentes y finalmente se estima la masa del cúmulo
17 suponiendo un cúmulo esférico.
18
19 Autora: Irati Lizaso Berrueta
20 """
21
22 import numpy as np
23 import matplotlib.pyplot as plt
24 from matplotlib.colors import LogNorm
25 import astropy.units as u
26 from astropy.constants import G,c
```

```

27 from lenstools.simulations import PotentialPlane, RayTracer
28 from astropy.io import fits
29 from astropy.cosmology import Planck15
30 import time
31 from skimage.measure import shannon_entropy
32 import scipy as scp
33
34 start = time.time()
35
36
37     ##Funciones:
38
39 def fx_densidad(x):
40     """
41     Recibiendo la coordenada radial adimensional x (tanto escalar
42     como vector), esta función retorna el valor de la función F(x)
43     necesaria para calcular la densidad de masa superficial.
44     Cuando la coordenada radial es nula, la función retorna un valor
45     nulo para evitar divergencias.
46
47     Parámetros:
48     -----
49     x : coordenada radial adimensional.
50
51     Retorna:
52     -----
53     F(x), valor de la función en x, definida a trozos.
54     """
55     if np.isscalar(x):
56         if x > 1.0:
57             return (1/(x**2-1))*(1-(np.arccos(1/x)/(np.sqrt(x**2-1))))
58         elif x == 1.0:
59             return 1/3
60         elif x == 0.0:
61             return 0 # para evitar que la función diverja
62         else:
63             return (1/(x**2-1))*(1-(np.arccosh(1/x)/
64                                     (np.sqrt(1-x**2))))
65             #arccosh(1/0) tiene limite infinito
66     else:
67         y = np.zeros(x.shape)
68         y[x>1] = (1/(x[x>1]**2-1))*(1-(np.arccos(1/x[x>1])/
69                                     (np.sqrt(x[x>1]**2-1))))
70         y[x==1] = 1/3
71         y[x==0] = 0.0
72         return y
73
74 def densidadNFW(x,y, x0,y0, rs, epsilon, orientacion,c_par, n=512):
75     """
76     Función que calcula la densidad de masa superficial para una
77     distribución elíptica de NFW.
78

```

```

79     Parámetros:
80     -----
81     x : Coordenada x horizontal del plano.
82     y : Coordenada y vertical del plano.
83     x0 : Centro de la coordenada horizontal.
84     y0 : Centro de la coordenada vertical.
85     rs : radio de escala.
86     epsilon : excentricidad de la elipse.
87     orientacion : orientación de la elipse.
88     c_par : parámetro de concentración del cúmulo.
89     n (OPCIONAL): dimensión del mapa de densidad (nxn).
90                     Por defecto, n=512.
91
92     Retorna:
93     -----
94     Valor de la densidad de masa superficial para una distribución
95     elíptica de NFW.
96     """
97     r = np.sqrt(np.sqrt(np.sqrt(1-epsilon**2))*(x-x0)**2+
98                 np.sqrt(1/np.sqrt(1-epsilon**2))*(y-y0)**2-
99                 orientacion*((x-x0)*(y-y0)))
100     dens_c = 200*c_par**3/(3*np.log(1+c_par)-c_par/(1+c_par))
101
102     return dens_c*(2*3*69*(u.km/u.s/u.Mpc)*69*(u.km/u.s/u.Mpc)/
103                (8*np.pi*G)*rs*u.Mpc*fx_densidad(r/rs))
104
105 def mapa_densidad(x0,y0, rs, epsilon, orientacion, c_par, n=512,
106                  dibujar=True):
107     """
108     Esta función crea una matriz de nxn píxeles con el valor de la
109     densidad superficial de masa en cada píxel. Dibuja esta densidad
110     junto con la distribución de contornos, en escalas normal y
111     logarítmica, a no ser que dibujar = False.
112     Como el valor central de la densidad (r = 0) tiende a infinito,
113     y la función fx_densidad(x) lo inicializa a 0, se le asigna el
114     valor medio de sus vecinos.
115
116     Parámetros:
117     -----
118     x0 : Centro de la coordenada horizontal.
119     y0 : Centro de la coordenada vertical.
120     rs : radio de escala.
121     epsilon : excentricidad de la elipse.
122     orientacion : orientación de la elipse.
123     c_par : parámetro de concentración del cúmulo.
124     n (OPCIONAL): dimensión del mapa de densidad (nxn).
125                     Por defecto, n=512.
126     dibujar (OPCIONAL): bool, dibuja la densidad y la distribución
127                          de contornos en escalas normal y logarítmica
128                          si dibujar = True.
129                          Por defecto, dibujar = True.
130

```

```

131     Retorna:
132     -----
133     Una matriz (nxn) de la densidad superficial de masa.
134     """
135     y = np.arange(0,n,1,float)
136     x = y[:,np.newaxis]
137     # como tras la maximización los centros son de tipo float:
138     x0 = int(x0)
139     y0 = int(y0)
140     imagen = (densidadNFW(x,y, x0,y0,rs,epsilon,orientacion,c_par,n)
141               *u.m*u.m/u.kg)
142     #Para corregir el valor que se va a infinito:
143     imagen[x0,y0] = (imagen[x0-1,y0]+imagen[x0+1,y0])/2
144     #Dibujo el mapa de densidad y su contorno.
145     if dibujar==True:
146         fig,axs = plt.subplots(2,2,figsize=(12,12))
147         fig.colorbar(axs[0,0].imshow(imagen, origin='lower',
148                                     cmap='viridis'), ax=axs[0,0])
149         fig.colorbar(axs[1,0].imshow(np.log(imagen), origin='lower',
150                                     cmap='viridis'), ax=axs[1,0])
151         axs[0,0].imshow(imagen, origin='lower',
152                         cmap='viridis')
153         axs[0,1].contour(imagen, origin='lower',
154                         cmap='viridis')
155         axs[1,0].imshow(np.log(imagen), origin='lower',
156                         cmap='viridis')
157         axs[1,1].contour(np.log(imagen), origin='lower',
158                         cmap='viridis')
159
160     return imagen
161
162 def hx_potencial(x):
163     """
164     Recibiendo la coordenada radial adimensional x (tanto escalar
165     como vector), esta función retorna el valor de la función h(x)
166     necesaria para calcular el potencial de lente.
167     Cuando la coordenada radial es nula, la función retorna un valor
168     nulo para evitar divergencias.
169
170     Parámetros:
171     -----
172     x : coordenada radial adimensional.
173
174     Retorna:
175     -----
176     h(x), valor de la función en x, definida a trozos.
177     """
178     if np.isscalar(x):
179         if x == 0.0:
180             return 0 # para evitar que la función diverja
181         elif x > 1.0:
182             return np.log(x/2)**2+np.arccos(1/x)**2

```

```

183         #log0 y arccosh(1/0) !!
184     else:
185         return np.log(x/2)**2+np.arccosh(1/x)**2
186 else:
187     y = np.zeros(x.shape)
188     y[x==0] = 1.0
189     y[x>1] = np.log(x[x>1]/2)**2+np.arccos(1/x[x>1])**2
190     return y
191
192 def potencial(x,y, x0, y0, rs, epsilon, orientacion, c_par, n=512):
193     """
194     Función que calcula el potencial de lensing para una distribución
195     elíptica de NFW.
196
197     Parámetros:
198     -----
199     x : Coordenada x horizontal del plano.
200     y : Coordenada y vertical del plano.
201     x0 : Centro de la coordenada horizontal.
202     y0 : Centro de la coordenada vertical.
203     rs : radio de escala.
204     epsilon : excentricidad de la elipse.
205     orientacion : orientación de la elipse.
206     c_par : parámetro de concentración del cúmulo.
207     n (OPCIONAL): dimensión del mapa de densidad (nxn).
208                   Por defecto, n = 512.
209
210     Retorna:
211     -----
212     Valor del potencial de lensing para una distribución elíptica
213     de NFW.
214     """
215     r = np.sqrt(np.sqrt(np.sqrt(1-epsilon**2))*(x-x0)**2+
216                 np.sqrt(1/np.sqrt(1-epsilon**2))*(y-y0)**2-
217                 orientacion*((x-x0)*(y-y0)))
218     dens_c = 200*c_par**3/(3*np.log(1+c_par)-c_par/(1+c_par))
219
220     return (3*dens_c*(69*(u.km/u.s/u.Mpc))**2*rs*u.Mpc*
221            Planck15.angular_diameter_distance(0.1)*
222            Planck15.angular_diameter_distance(0.4)/
223            (Planck15.angular_diameter_distance(0.5)*c*c)*
224            hx_potencial(r/rs))
225
226 def potencial_a_fits(imagen_pot,nombre_archivo, z):
227     """
228     Función que crea un archivo fits con el mapa de potencial y
229     modifica su header para cumplir los requerimientos de LensTools
230     para poder leerlo adecuadamente.
231
232     Parámetros:
233     -----
234     imagen_pot : matriz nxn con los valores del potencial.

```

```

235     nombre_archivo : string, nombre del archivo guardado.
236     z : redshift del plano de lente.
237
238     Retorna:
239     -----
240     Nada.
241     """
242     #Calculo la distancia comóvil a partir del redshift:
243     comoving_distance = float(Planck15.comoving_distance(z) / u.Mpc)
244     # Creo el archivo:
245     imagen_pot = np.array(imagen_pot)
246     hdu = fits.PrimaryHDU(imagen_pot)
247     hdu.writeto(nombre_archivo, overwrite=True)
248     # Para actualizar el header a los requerimientos de LensTools:
249     hdul=fits.open(nombre_archivo)
250     hdr = hdul[0].header
251     hdr.append(('H0', 67.66, 'Hubble constant in km/s*Mpc'), end=True)
252     hdr.append(('H', 0.6766, 'Dimensionless Hubble constant'),
253               end=True)
254     hdr.append(('OMEGA_M', 0.3111, 'Dark Matter density'), end=True)
255     hdr.append(('OMEGA_L', 0.6889, 'Dark Energy density'), end=True)
256     hdr.append(('WO', -1.0, 'Dark Energy equation of state'),
257               end=True)
258     hdr.append(('WA', 0.0, 'Dark Energy running equation of state'),
259               end=True)
260     hdr.append(('Z', z, 'Redshift of the lens plane'), end=True)
261     hdr.append(('CHI', comoving_distance,
262               'Comoving distance in Mpc/h'), end=True)
263     hdr.append(('ANGLE', 3.5, 'Side angle in degrees'), end=True)
264     hdr.append(('NPART', 14929298.0,
265               'Number of particles on the plane'), end=True)
266     hdul.close()
267     hdu = fits.PrimaryHDU(imagen_pot, header = hdr)
268     hdu.writeto(nombre_archivo, overwrite=True)
269
270 def mapa_potencial(x0,y0,rs,epsilon,orientacion,c_par,nombre_archivo,
271                   z, n=512, dibujar=True, fits=True,densidad=True):
272     """
273     Esta función crea una matriz de nxn píxeles con el valor del
274     potencial de lensing en cada píxel. Dibuja este potencial a no
275     ser que dibujar == False.
276     Como el valor central de la densidad (r = 0) tiende a infinito,
277     y la función hx_potencial(x) lo inicializa el centro a 0,
278     se le asigna el valor medio de sus vecinos verticales.
279     Adicionalmente, crea un archivo fits llamando a la función
280     potencial_a_fits, si fits == True.
281
282     Parámetros:
283     -----
284     x0 : Centro de la coordenada horizontal.
285     y0 : Centro de la coordenada vertical.
286     rs : radio de escala.

```

```

287     epsilon : excentricidad de la elipse.
288     orientacion : orientación de la elipse.
289     c_par : parámetro de concentración del cúmulo.
290     nombre_archivo : string, nombre con el que se crea el archivo.
291     z : redshift del plano de la lente.
292     n (OPCIONAL): Número de píxeles por cada lado.
293                     Por defecto, n=512.
294     dibujar (OPCIONAL): bool, dibuja la densidad y la distribución
295                         de contornos si es dibujar = True.
296                         Por defecto, dibujar == True.
297     fits (OPCIONAL): bool, crea un archivo fits con nombre_archivo
298                     si es fits = True.
299                     Por defecto, fits == True.
300
301     Retorna:
302     -----
303     La matriz (nxn) del potencial de lensing.
304     """
305     y = np.arange(0,n,1,float)
306     x = y[:,np.newaxis]
307     x0 = int(x0)
308     y0 = int(y0)
309     imagen_pot = potencial(x,y, x0,y0,rs,epsilon,orientacion,c_par,n)
310     imagen_pot[x0,y0] = (imagen_pot[x0-1,y0]+imagen_pot[x0+1,y0])/2
311     #Dibujo el mapa de potencial.
312     if dibujar == True:
313         plt.figure(figsize=(5,5))
314         plt.imshow(imagen_pot, origin='lower', cmap='viridis')
315         plt.colorbar()
316     #Llamo a la función potencial_a_fits y convierto este mapa a
317     #un archivo FITS.
318     if fits == True:
319         potencial_a_fits(imagen_pot, nombre_archivo, z)
320
321     return imagen_pot
322
323 def elipses(seed, n=512):
324     '''
325     Esta función genera una imagen nxn de elipses aleatorias.
326
327     Parámetros:
328     -----
329     seed : INT, generador de aleatoriedad.
330     n (OPCIONAL): Dimensión de la imagen.
331                     Por defecto, n=512.
332
333     Retorna:
334     -----
335     La matriz de elipses aleatorias.
336     '''
337     np.random.seed(seed)
338     elipses = np.zeros((n,n))

```

```

339     for eli in range(120):
340         epsilon = np.random.rand()
341         x0 = np.random.rand()*n
342         y0 = np.random.rand()*n
343         orientacion = np.pi*np.random.rand() - np.pi/2
344         radio = np.random.rand()*25
345
346         y = np.arange(0,n,1,float)
347         x = y[:,np.newaxis]
348         r = np.sqrt(np.sqrt(np.sqrt(1-epsilon**2))*(x-x0)**2+
349                     np.sqrt(1/(np.sqrt(1-epsilon**2)))*(y-y0)**2+
350                     orientacion*(x-x0)*(y-y0))
351
352         elipses = np.where(r>radio,elipses, np.random.rand()*5*
353                             np.exp(-(7.66925)*(r**(1/4)-1)))
354
355     return elipses
356
357 def scatter2imagen(scatter, magnitud):
358     '''
359     Función que convierte los datos de un scatterplot (necesario
360     para LensTools) en una imagen 512x512, siguiendo las
361     distribuciones de galaxias iniciales.
362
363     Parámetros:
364     -----
365     scatter : posiciones de los puntos del scatterplot.
366     magnitud : las distribuciones de galaxias.
367
368     Retorna:
369     -----
370     imagen : array de 2D de 512x512 píxeles.
371
372     '''
373     x_step, y_step = 36/512, 36/512
374     x = np.round(scatter[0]/u.deg / x_step).astype(int)
375     y = np.round(scatter[1]/u.deg / y_step).astype(int)
376     x_m, x_M = np.min(x), np.max(x)
377     y_m, y_M = np.min(y), np.max(y)
378     imagen = np.zeros((x_M - x_m + 1, y_M - y_m + 1),
379                       dtype=magnitud.dtype)
380
381     while imagen.shape[0]!=512 or imagen.shape[1]!=512:
382         #cambiarle el tamaño a 512:
383         x_step = x_step*imagen.shape[0]/512
384         y_step = y_step*imagen.shape[1]/512
385         x = np.round(scatter[0]/u.deg / x_step).astype(int)
386         y = np.round(scatter[1]/u.deg / y_step).astype(int)
387         x_m, x_M = np.min(x), np.max(x)
388         y_m, y_M = np.min(y), np.max(y)
389         imagen = np.zeros((x_M - x_m + 1, y_M - y_m + 1),
390                           dtype=magnitud.dtype) #se recalcula

```



```

391
392     imagen[x-x_m, y-y_m] = magnitud[np.where(magnitud>0)]
393     imagen = np.transpose(imagen)
394     return imagen
395
396 def entropia(p, imagen_lensada, imagen_original):
397     '''
398     Esta función crea un plano de potencial con los valores de los
399     parámetros introducidos en p, y deslensa la imagen lensada que
400     se le pasa como argumento. Después calcula la entropía
401     de la nueva imagen deslensada.
402
403     Parámetros:
404     -----
405     p : Array de dimensión (5,) con los parámetros del centro en x,
406         centro en y, radio de escala rs, excentricidad y orientación
407         del potencial.
408     imagen_lensada : imagen previamente lensada en formato LensTools.
409     imagen_original : imagen que se ha lensado, para utilizar los
410                     valores de sus píxeles.
411
412     Retorna:
413     -----
414     (-1)*H : la entropía de la nueva imagen deslensada.
415
416     '''
417     mapa_potencial(p[0],p[1],p[2],p[3],p[4],p[5],
418                   'potencial_prueba.fits',0.5, dibujar=False)
419     tracer = RayTracer(lens_mesh_size=512)
420     tracer.addLens(PotentialPlane.load('potencial_base.fits'))
421     tracer.addLens(PotentialPlane.load('potencial_prueba.fits'))
422     tracer.reorderLenses()
423     tracer.randomRoll(3)
424     imagen_delensada = tracer.shoot(imagen_lensada, 0.5)
425     H = shannon_entropy(scatter2imagen(imagen_delensada,
426                                       imagen_original))
427     return (-1)*H
428
429
430
431     ##Creo los planos lente:
432
433     #Llamo a la función mapa_potencial con los mismos valores que
434     #mapa_densidad para crear el mapa correspondiente.
435     mapa_potencial(256,256, 0.5, 0.7,2, 25, 'potencial_base.fits', 0.51,
436                   n=512,dibujar = False, fits = True)
437
438     #Creo el plano lente:
439     mapa_densidad(256, 256, 0.5, 0.7, -1.0, 25, dibujar = True)
440     mapa_potencial(256, 256, 0.5, 0.7, -1.0, 25, 'potencial.fits', 0.1,
441                   n=512,dibujar=True, fits = True)
442

```

```

443
444
445     ##Con los planos de lente creados, empiezo el RayTracing:
446
447 #Iniciación de la clase RayTracer:
448 tracer = RayTracer(lens_mesh_size=512)
449 tracer.addLens(PotentialPlane.load('potencial_base.fits'))
450 tracer.addLens(PotentialPlane.load('potencial.fits'))
451
452 tracer.reorderLenses()
453 tracer.randomRoll(3)
454
455
456
457     ##Lensing de un grid:
458
459 #Creo el grid:
460 s = np.linspace(0.0,3.5,16)
461 beta_G = np.array(np.meshgrid(s,s))*u.deg
462 #El lensing, obtenemos la imagen theta:
463 theta_G = tracer.shootForward(beta_G,z=0.5)
464 #Lo vuelvo a de-lensar:
465 beta_inv = tracer.shoot(theta_G,z=0.5)
466
467 #Dibujo el grid lensado y de-lensado:
468 fig,ax = plt.subplots(1,3,figsize=(18,5))
469 ax[0].scatter(beta_G[0],beta_G[1],color="black")
470 ax[0].set_title(r"Grid original $\beta_G$",fontsize=15)
471 ax[1].scatter(theta_G[0],theta_G[1],color="black")
472 ax[1].set_title(r"Grid aparente $\theta_G$",fontsize=15)
473 ax[2].scatter(beta_inv[0],beta_inv[1],color="black")
474 ax[2].set_title(r"Grid de-lensado $\beta_G$",fontsize=15)
475 for i in range(3):
476     ax[i].set_xlabel(r"$x_{%d}$".format(theta_G.unit.to_string()))
477     ax[i].set_ylabel(r"$y_{%d}$".format(theta_G.unit.to_string()))
478
479
480
481     ##Lensing de elipses aleatorias:
482
483 #Llamo a la función que me crea elipses aleatorias.
484 imagen = elipses(0, n=512)
485 # Ray tracing de las elipses.
486 beta_elipses=(np.array(np.where(imagen>0))*3.5/imagen.shape[0])*u.deg
487 beta_elipses = np.roll(beta_elipses,1,axis=0) #Para poder verlas bien.
488 theta_elipses = tracer.shootForward(beta_elipses, z=0.5)
489 beta_elipses_inv = tracer.shoot(theta_elipses, z=0.5)
490
491 #Dibujo la imagen (en array) de las elipses lensadas y de-lensadas.
492 fig,ax = plt.subplots(1,3,figsize=(18,5))
493 ax[0].imshow(scatter2imagen(beta_elipses, imagen)+0.2,
494             cmap = 'bone', norm = LogNorm(), origin = 'lower')

```

```

495 ax[1].imshow((scatter2imagen(theta_elipses, imagen)+0.2),
496               cmap = 'bone', norm = LogNorm(), origin = 'lower')
497 ax[2].imshow((scatter2imagen(beta_elipses_inv, imagen)+0.2),
498               cmap = 'bone', norm = LogNorm(), origin = 'lower')
499
500
501 #Estudio comparativo de las imágenes originales y de-lensadas.
502 original = scatter2imagen(beta_elipses, imagen)
503 delensado = scatter2imagen(beta_elipses_inv, imagen)
504 residuo = np.subtract(original, delensado)
505 plt.figure(figsize=(5,5))
506 plt.imshow(np.log(np.abs(residuo)+0.1), cmap = 'bone', origin='lower')
507
508
509
510     ##Estadística de la entropía para 4000 muestras.
511
512 lista_unlensed=[shannon_entropy(elipses(M,512)) for M in range(4000)]
513 lista_lensadas = [shannon_entropy
514                   (scatter2imagen(tracer.shootForward(np.roll(
515                       (np.array(np.where(elipses(M,512)>0)) *
516                       3.5/elipses(M,512).shape[0]) * u.deg,1,axis=0),
517                       z=0.5), elipses(M,512))) for M in range(4000)]
518 diferencia_entropias = np.subtract(lista_unlensed,lista_lensadas)
519
520 #Medias y desviaciones estándar:
521 media_originales = np.mean(lista_unlensed)
522 std_originales = np.std(lista_unlensed)
523 media_lensadas = np.mean(lista_lensadas)
524 std_lensadas = np.std(lista_lensadas)
525 media_residuo = np.mean(diferencia_entropias)
526 std_residuo = np.std(diferencia_entropias)
527
528 # Dibujo los histogramas:
529 fig,ax = plt.subplots(1,3,figsize=(18,5))
530 ax[0].hist(lista_unlensed, bins='scott')
531 # ax[0].set_title('Entropía de las galaxias originales')
532 ax[1].hist(lista_lensadas, bins='scott')
533 # ax[1].set_title('Entropía de las galaxias lensadas')
534 ax[2].hist(diferencia_entropias, bins='scott')
535 # ax[2].set_title('Entropía del residuo')
536 for i in range(3):
537     ax[i].set_xlabel(r'H')
538     ax[i].set_ylabel(r'Frecuencia')
539
540
541
542     ##Maximización de la aleatoriedad.
543
544     #p=[x0,y0,rs,epsilon,orientacion,c_par]
545
546 # Lenso las galaxias de fondo con un potencial con parámetros:

```

```

547         #p=[256,256,0.50,0.7,-1.0,25]
548 imagen = elipses(0)
549 imagen_original = (np.array(np.where(imagen>0)) *
550                     3.5/imagen.shape[0]) * u.deg
551 imagen_original = np.roll(imagen_original,1,axis=0)
552 imagen_lensada = tracer.shootForward(imagen_original, 0.5)
553 imagen_delensada = tracer.shoot(imagen_lensada, 0.5)
554 p0 = np.array([256, 256, 0.50, 0.7, -1.0, 25])
555
556 # Maximizo la entropía:
557 maxima_entropia = scp.optimize.minimize(entropia,p0,args=
558                                         (imagen_lensada,elipses(0)),method='trust-constr',
559                                         bounds=((254, 258), (254, 258), (0.1,0.5),
560                                                  (0.6,0.9), (-np.pi/2,np.pi/2), (15,25)))
561 estimado=[maxima_entropia.x[0],maxima_entropia.x[1],maxima_entropia.
562            x[2], maxima_entropia.x[3], maxima_entropia.x[4],
563            maxima_entropia.x[5]]
564
565 # Calculo las densidades superficiales de masa real y estimada.
566 densidad_usada = mapa_densidad(p0[0], p0[1], p0[2], p0[3], p0[4],
567                                p0[5], dibujar=False)
568 densidad_estimada=mapa_densidad(estimado[0],estimado[1],estimado[2],
569                                estimado[3],estimado[4], estimado[5],
570                                dibujar=False)
571
572 # Las dibujo para poder comparar:
573 fig,ax = plt.subplots(1,2,figsize=(12,6))
574 im =ax[0].imshow(np.log(densidad_usada),origin='lower',cmap='viridis')
575 ax[1].imshow(np.log(densidad_estimada), origin='lower',cmap='viridis')
576 fig.colorbar(im, ax=ax.ravel().tolist(), shrink=0.95)
577
578 # Ahora mantengo la misma imagen inicial, pero distintas lentes.
579 tabla = np.zeros((3*3*5*4*4*10,6,2))
580 it = 0
581 for dx in range(3):#3
582     for dy in range(3):#3
583         for dr in range(5):#5
584             for deps in range(4):#4
585                 for dor in range(4):#4
586                     for dc in range(10):#10
587                         mapa_potencial(255+dx,255+dy, 0.1+0.1*dr,
588                                         0.6+0.1*deps,-np.pi/2+dor,
589                                         15+dc,'potencial_prueba.fits',
590                                         0.5, dibujar=False)
591                         tracer = RayTracer(lens_mesh_size=512)
592                         tracer.addLens(PotentialPlane.load
593                                       ('potencial_base.fits'))
594                         tracer.addLens(PotentialPlane.load
595                                       ('potencial_prueba.fits'))
596                         tracer.reorderLenses()
597                         tracer.randomRoll(3)
598

```

```

599     imagen = elipses(0)
600     imagen_original=(np.array(np.where(imagen>0))
601                             *3.5/imagen.shape[0]) * u.deg
602     imagen_original =np.roll(imagen_original,
603                               1,axis=0)
604     imagen_lensada=tracer.shootForward(
605                             imagen_original,0.5)
606     imagen_delensada=tracer.shoot(
607                             imagen_lensada,0.5)
608     p0 = np.array([255+dx,255+dy, 0.1+0.1*dr,
609                   0.6+0.1*deps,-np.pi/2+dor,
610                   15+dc])
611     maxima_entropia=scp.optimize.minimize(entropia
612                                           ,p0,args=(imagen_lensada,imagen),
613                                           method='trust-constr',bounds=(
614                                           (254, 258),(254, 258),(0.1,0.5),
615                                           (0.6,0.9), (-np.pi/2,np.pi/2),
616                                           (15,25)))
617     #Guardo los datos .x para compararlos:
618     tabla[it][:,0]= p0
619     tabla[it][:,1]= [maxima_entropia.x[0],
620                     maxima_entropia.x[1],
621                     maxima_entropia.x[2],
622                     maxima_entropia.x[3],
623                     maxima_entropia.x[4],
624                     maxima_entropia.x[5]]
625     it +=1
626
627 #Importo la tabla con los potenciales reales y estimados para
628 # no tener que iterar otra vez.
629 tabla = np.load('tabla_parametros.npy')
630 diferencias = tabla[:,0]-tabla[:,1]
631
632 #La media y la desviación de las diferencias entre real y estimado:
633 x0_dif_media = np.mean(diferencias[:,0])
634 y0_dif_media = np.mean(diferencias[:,1])
635 rs_dif_media = np.mean(diferencias[:,2])
636 epsilon_dif_media = np.mean(diferencias[:,3])
637 orient_dif_media = np.mean(diferencias[:,4])
638 c_dif_media = np.mean(diferencias[:,5])
639 x0_dif_std = np.std(diferencias[:,0])
640 y0_dif_std = np.std(diferencias[:,1])
641 rs_dif_std = np.std(diferencias[:,2])
642 epsilon_dif_std = np.std(diferencias[:,3])
643 orient_dif_std = np.std(diferencias[:,4])
644 c_dif_std = np.std(diferencias[:,5])
645
646 # Dibujo los histogramas:
647 fig, axs = plt.subplots(2, 3, figsize=(12,8))
648 axs[0, 0].hist(diferencias[:,0], bins='scott')
649 axs[0, 0].set_title(r"Centro $x_0$")
650 axs[0, 1].hist(diferencias[:,1], bins='scott')

```

```

651 axs[0, 1].set_title(r'Centro $y_0$')
652 axs[0, 2].hist(diferencias[:,2], bins='scott')
653 axs[0, 2].set_title(r'Radio de escala $r_s$')
654 axs[1, 0].hist(diferencias[:,3], bins='scott')
655 axs[1, 0].set_title(r"Excentricidad $\varepsilon$")
656 axs[1, 1].hist(diferencias[:,4], bins='scott')
657 axs[1, 1].set_title(r'Orientación $\phi$')
658 axs[1, 2].hist(diferencias[:,5], bins='scott')
659 axs[1, 2].set_title(r'Parámetro de concentración $c$')
660
661
662
663     ##Masas estimadas para un modelo esférico:
664
665 #Masa dentro de un radio de 4Mpc para c=25 y rs=0.5 Mpc:
666 masa_grande = (4*np.pi*200/3*25*25*25/(np.log(26)-25/26)*3*69*
667               (u.km/u.s/u.Mpc)*69*(u.km/u.s/u.Mpc)/(8*np.pi*G)*0.5*
668               u.Mpc*0.5*u.Mpc*0.5*u.Mpc*(np.log(0.5+4/0.5)+0.5/4.5-1))
669 print(f'Grande:{masa_grande.to(u.kg)/(1.98847e30*u.kg)} masas solares'
670       )
671
672 #Masa dentro de un radio de 4Mpc para c=15 y rs=0.1 Mpc:
673 masa_pequeña = (4*np.pi*200/3*15*15*15/(np.log(16)-15/16)*3*69*
674                (u.km/u.s/u.Mpc)*69*(u.km/u.s/u.Mpc)/(8*np.pi*G)*0.1*
675                u.Mpc*0.1*u.Mpc*0.1*u.Mpc*(np.log(0.1+4/0.1)+0.1/4.1-1))
676 print(f'Pequeño:{masa_pequeña.to(u.kg)/(1.98847e30*u.kg)} masas
677       solares')
678
679 #Masa dentro de un radio de 4Mpc para c estimado y rs estimado:
680 masa_est = (4*np.pi*200/3*24.5097*24.5097*24.5097/(np.log(24.5097+1)
681              -24.5097/25.5097)*3*69*(u.km/u.s/u.Mpc)*69*(u.km/u.s/u.Mpc)/
682              (8*np.pi*G)*0.2840*u.Mpc*0.2840*u.Mpc*0.2840*u.Mpc*
683              (np.log(0.2840+4/0.2840)+0.2840/4.2840-1))
684 print(f'Estimado:{masa_est.to(u.kg)/(1.98847e30*u.kg)} masas solares')
685
686
687     ## Finalmente calculo el tiempo que tarda el programa en
688     ## ejecutarlo todo para estudiar la eficiencia del código:
689
690 tiempo_total = (time.time() - start)
691 print('')
692 print(f'Tiempo requerido en total: {tiempo_total/60} minutos')

```

Bibliografía

- [1] S. W. Allen, A. E. Evrard, A. B. Mantz, [Cosmological parameters from observations of galaxy clusters](#), Annual Review of Astronomy and Astrophysics 49 (1) (2011) 409–470. [doi:10.1146/annurev-astro-081710-102514](#).
URL <http://dx.doi.org/10.1146/annurev-astro-081710-102514>
- [2] G. Bertone, D. Hooper, History of dark matter, Reviews of Modern Physics 90 (4) (Oct 2018). [doi:10.1103/revmodphys.90.045002](#).
- [3] J.-P. Luminet, The dark matter enigma (2021). [arXiv:2101.10127](#).
- [4] N. Kaiser, G. Squires, Mapping the Dark Matter with Weak Gravitational Lensing, 404 (1993) 441. [doi:10.1086/172297](#).
- [5] P. Schneider, C. Seitz, Steps towards nonlinear cluster inversion through gravitational distortions. I. Basic considerations and circular clusters., 294 (1995) 411–431. [arXiv:astro-ph/9407032](#).
- [6] S. Wallington, C. S. Kochanek, R. Narayan, [Lensmem: A gravitational lens inversion algorithm using the maximum entropy method](#), The Astrophysical Journal 465 (1996) 64. [doi:10.1086/177401](#).
URL <http://dx.doi.org/10.1086/177401>
- [7] S. L. Bridle, M. P. Hobson, A. N. Lasenby, R. Saunders, [A maximum-entropy method for reconstructing the projected mass distribution of gravitational lenses](#), Monthly Notices of the Royal Astronomical Society 299 (3) (1998) 895–903. [doi:10.1046/j.1365-8711.1998.01877.x](#).
URL <http://dx.doi.org/10.1046/j.1365-8711.1998.01877.x>
- [8] S. Seitz, P. Schneider, M. Bartelmann, Entropy-regularized maximum-likelihood cluster mass reconstruction, 337 (1998) 325–337. [arXiv:astro-ph/9803038](#).
- [9] S. Bridle, M. Hobson, R. Saunders, A. Lasenby, Maximum-entropy reconstruction of gravitational lenses using shear and/or magnification data (11 2000).
- [10] S. L. Jaki, Johann Georg von Soldner and the Gravitational Bending of Light, with an english translation of his essay on it published in 1801, Foundations of Physics 8 (11-12) (1978) 927–950. [doi:10.1007/BF00715064](#).

- [11] D. Valls-Gabaud, The conceptual origins of gravitational lensing, AIP Conference Proceedings (2006). [doi:10.1063/1.2399715](https://doi.org/10.1063/1.2399715).
- [12] R. Narayan, M. B. CfA, Mpa, Lectures on gravitational lensing, 1996.
- [13] E. Zackrisson, T. Riehm, Gravitational lensing as a probe of cold dark matter subhalos, *Advances in Astronomy 2010* (2010) 1–14. [doi:10.1155/2010/478910](https://doi.org/10.1155/2010/478910).
- [14] T. Treu, P. J. Marshall, D. Clowe, Resource Letter GL-1: Gravitational Lensing, *American Journal of Physics* 80 (9) (2012) 753–763. [doi:10.1119/1.4726204](https://doi.org/10.1119/1.4726204).
- [15] R. D. Blandford, R. Narayan, Cosmological applications of gravitational lensing., 30 (1992) 311–358. [doi:10.1146/annurev.astro.30.1.311](https://doi.org/10.1146/annurev.astro.30.1.311).
- [16] Paraficz, D., Hjorth, J., Gravitational lenses as cosmic rulers: Ω_m , Ω_Λ from time delays and velocity dispersions, *A&A* 507 (3) (2009) L49–L52. [doi:10.1051/0004-6361/200913307](https://doi.org/10.1051/0004-6361/200913307).
- [17] P. Schneider, C. Kochanek, J. Wambsganss, *Gravitational Lensing: Strong, Weak and Micro*, 2006. [doi:10.1007/978-3-540-30310-7](https://doi.org/10.1007/978-3-540-30310-7).
- [18] A. Petri, Mocking the weak lensing universe: The LensTools Python computing package, *Astronomy and Computing* 17 (2016) 73–79. [arXiv:1606.01903](https://arxiv.org/abs/1606.01903), [doi:10.1016/j.ascom.2016.06.001](https://doi.org/10.1016/j.ascom.2016.06.001).
- [19] J. F. Navarro, C. S. Frenk, S. D. M. White, A universal density profile from hierarchical clustering, *The Astrophysical Journal* 490 (2) (1997) 493–508. [doi:10.1086/304888](https://doi.org/10.1086/304888).
- [20] R. E. Sanderson, J. Hartke, A. Helmi, [Modeling the gravitational potential of a cosmological dark matter halo with stellar streams](https://arxiv.org/abs/1703.03215), *The Astrophysical Journal* 836 (2) (2017) 234. [doi:10.3847/1538-4357/aa5eb4](https://doi.org/10.3847/1538-4357/aa5eb4). URL <https://doi.org/10.3847/1538-4357/aa5eb4>
- [21] J. F. Navarro, C. S. Frenk, S. D. M. White, Simulations of X-ray clusters, 275 (3) (1995) 720–740. [arXiv:astro-ph/9408069](https://arxiv.org/abs/astro-ph/9408069), [doi:10.1093/mnras/275.3.720](https://doi.org/10.1093/mnras/275.3.720).
- [22] J. F. Navarro, C. S. Frenk, S. D. M. White, The structure of cold dark matter halos, *The Astrophysical Journal* 462 (1996) 563. [doi:10.1086/177173](https://doi.org/10.1086/177173).
- [23] M. Bartelmann, Arcs from a universal dark-matter halo profile., 313 (1996) 697–702. [arXiv:astro-ph/9602053](https://arxiv.org/abs/astro-ph/9602053).
- [24] R. Barkana, Fast calculation of a family of elliptical gravitational lens models, *The Astrophysical Journal* 502 (2) (1998) 531–537. [doi:10.1086/305950](https://doi.org/10.1086/305950).

- [25] G. Golse, J.-P. Kneib, Pseudo elliptical lensing mass model: Application to the NFW mass distribution, *Astronomy Astrophysics* 390 (3) (2002) 821–827. [doi:10.1051/0004-6361:20020639](https://doi.org/10.1051/0004-6361:20020639).
- [26] S. Floor, A. Melott, C. Miller, G. Bryan, Eccentricity evolution in simulated galaxy clusters, *The Astrophysical Journal* 591 (01 2003). [doi:10.1086/375493](https://doi.org/10.1086/375493).
- [27] Y. P. Jing, [The density profile of equilibrium and nonequilibrium dark matter halos](https://doi.org/10.1086/308809), *The Astrophysical Journal* 535 (1) (2000) 30–36. [doi:10.1086/308809](https://doi.org/10.1086/308809).
URL <http://dx.doi.org/10.1086/308809>
- [28] A. Mazure, H. V. Capelato, Exact solutions for the spatial de Vaucouleurs and Sérsic laws and related quantities, *Astronomy Astrophysics* 383 (2) (2002) 384–389. [doi:10.1051/0004-6361:20011751](https://doi.org/10.1051/0004-6361:20011751).
- [29] I. Trujillo, A. Graham, N. Caon, On the estimation of galaxy structural parameters: the Sérsic model, *Monthly Notices of the Royal Astronomical Society* 326 (2001) 869–876.
- [30] N. Aghanim, Y. Akrami, M. Ashdown, J. Aumont, C. Baccigalupi, M. Ballardini, A. J. Banday, R. B. Barreiro, N. Bartolo, et al., Planck 2018 results, *Astronomy Astrophysics* 641 (2020) A6. [doi:10.1051/0004-6361/201833910](https://doi.org/10.1051/0004-6361/201833910).
- [31] C. E. Shannon, A mathematical theory of communication, *Bell System Technical Journal* 27 (3) (1948) 379–423. [doi:https://doi.org/10.1002/j.1538-7305.1948.tb01338.x](https://doi.org/10.1002/j.1538-7305.1948.tb01338.x).
- [32] Y. Wu, Y. Zhou, G. Saveriades, S. Agaian, J. P. Noonan, P. Natarajan, Local Shannon entropy measure with statistical tests for image randomness, *Information Sciences* 222 (2013) 323–342, including Special Section on New Trends in Ambient Intelligence and Bio-inspired Systems. [doi:https://doi.org/10.1016/j.ins.2012.07.049](https://doi.org/10.1016/j.ins.2012.07.049).
- [33] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, scikit-image: image processing in Python, *PeerJ* 2 (2014) e453.
- [34] E. Jones, T. Oliphant, P. Peterson, et al., [SciPy: Open source scientific tools for Python](https://doi.org/10.1002/j.1538-7305.1999.tb00809.x) (2001–).
URL <http://www.scipy.org/>
- [35] M. Olamaie, M. P. Hobson, K. J. B. Grainge, A simple parametric model for spherical galaxy clusters, *Monthly Notices of the Royal Astronomical Society* 423 (2) (2012) 1534–1543. [doi:10.1111/j.1365-2966.2012.20980.x](https://doi.org/10.1111/j.1365-2966.2012.20980.x).
- [36] C. Balland, A. Blanchard, On the uncertainty in X-ray cluster mass estimates (10 1995). [arXiv:astro-ph/9510130](https://arxiv.org/abs/astro-ph/9510130).

- [37] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. C. et al., Array programming with NumPy, *Nature* 585 (7825) (2020) 357–362. [doi:10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2).
- [38] J. D. Hunter, Matplotlib: A 2d graphics environment, *Computing in Science & Engineering* 9 (3) (2007) 90–95. [doi:10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55).
- [39] Astropy Collaboration, The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package, 156 (3) (2018) 123. [arXiv:1801.02634](https://arxiv.org/abs/1801.02634), [doi:10.3847/1538-3881/aabc4f](https://doi.org/10.3847/1538-3881/aabc4f).
- [40] Astropy Collaboration, Astropy: A community Python package for astronomy, 558 (2013) A33. [arXiv:1307.6212](https://arxiv.org/abs/1307.6212), [doi:10.1051/0004-6361/201322068](https://doi.org/10.1051/0004-6361/201322068).